THE UNIVERSITY *of* EDINBURGH

Low-Power Adaptive Control Scheme Using Switching Activity Measurement Method for Reconfigurable Analog-to-Digital Converters

Mohd Zulhakimi Ab Razak

Thesis submitted for the degree of
Doctor of Philosophy

The University of Edinburgh
United Kingdom
November 2013

Dedicated

In the loving memory of my mother, my idol and inspiration, and the best friend I have ever had

To my father, whose courage, and understanding have made me who I am today

To my wife, whose love, patience, and support have driven me to complete this thesis

Declaration of Originality

I hereby declare that the research recorded in this thesis and the thesis itself was composed and originated entirely by myself in the School of Engineering at The University of Edinburgh, United Kingdom, except when otherwise stated.

Mohd Zulhakimi Ab Razak

Edinburgh, Scotland

November 2013

Low-Power Adaptive Control Scheme Using Switching Activity Measurement Method for Reconfigurable Analog-to-Digital Converters

A thesis submitted to the University of Edinburgh in candidature
for the degree of Doctor of Philosophy

November 2013

by

Mohd. Zulhakimi Ab. Razak

Abstract

Power consumption is a critical issue for portable devices. The ever-increasing demand for multimode wireless applications and the growing concerns towards power-aware green technology make dynamically reconfigurable hardware an attractive solution for overcoming the power issue. This is due to its advantages of flexibility, reusability, and adaptability. During the last decade, reconfigurable analog-to-digital converters (ReADCs) have been used to support multimode wireless applications. With the ability to adaptively scale the power consumption according to different operation modes, reconfigurable devices utilise the power supply efficiently. This can prolong battery life and reduce unnecessary heat emission to the environment. However, current adaptive mechanisms for ReADCs rely upon external control signals generated using digital signal processors (DSPs) in the baseband. This thesis aims to provide a single-chip solution for real-time and low-power ReADC implementations that can adaptively change the converter resolution according to signal variations without the need of the baseband processing. Specifically, the thesis focuses on the analysis, design and implementation of a low-power digital controller unit for ReADCs. In this study, the following two

important reconfigurability issues are investigated: i) the detection mechanism for an adaptive implementation, and ii) the measure of power and area overheads that are introduced by the adaptive control modules.

This thesis outlines four main achievements to address these issues. The first achievement is the development of the switching activity measurement (SWAM) method to detect different signal components based upon the observation of the output of an ADC. The second achievement is a proposed adaptive algorithm for ReADCs to dynamically adjust the resolution depending upon the variations in the input signal. The third achievement is an ASIC implementation of the adaptive control module for ReADCs. The module achieves low reconfiguration overheads in terms of area and power compared with the main analog part of a ReADC. The fourth achievement is the development of a low-power noise detection module using a conventional ADC for signal improvement. Taken together, the findings from this study demonstrate the potential use of switching activity information of an ADC to adaptively control the circuits, and simultaneously expanding the functionality of the ADC in electronic systems.

Acknowledgement

Praise be to Allah Almighty, my Creator and Sustainer, who made me capable enough to complete this thesis.

I cannot begin to thank the countless people who have influenced my life and education. It would not have been possible to write this thesis without the help and support from various individuals, to only some of whom it is possible to give particular mention here.

I would like to thank my supervisor, Professor Tughrul Arslan for his professionalism, technical expertise, and support throughout my doctoral study. I also would like to thank my second supervisor, Dr. Alister Hamilton, who helped me during the early stage of the study.

I would like to express my deepest gratitude to Dr. Ahmet Teyfik Erdogan for many things. His mentoring commitment, kindness, help, and advice during the different stages of my PhD years have been wonderful. His technical acumen, precise suggestions, and lively discussions are heartily valued.

My thanks extended to Prof. Dr.-Ing. Herbert Thurner and his assistant Mr. Kurt Fischer, both at the University of Applied Science Rosenheim, whom have initially sparked, encouraged, and driven my interest in RF, analog, and digital circuits design.

I would like to thank and dedicate this thesis to all my teachers, lecturers, mentors, and gurus throughout my life and my academic years.

Dr. Wim Vanderbauwhede (University of Glasgow, UK) and Dr. Khaled Benkrid (University of Edinburgh, UK) examined this thesis. Their comments

and suggestions surely had improved the thesis. Special thanks go to both.

Also, to all my colleagues in the SLI group, previous and current members, for timely discussion, help, and moral support throughout my PhD. Thank you for all those memorable years.

Special thanks to the University of Malaya and the Ministry of Higher Education of Malaysia for the generous funding and scholarship.

My acknowledgement also goes to all my friends. All the encouragement, cooperation, understanding, and endless friendship will always be remembered.

Last but not least, I would like to express my heartfelt gratitude to Mrs. Diane Reid for her kind assistance and help, especially on the formalities regarding the study.

Above all, I wish to express my special appreciation to my wife, Fazia Adyani Ahmad Fuad, for her care, untiring love, and great patience during this doctoral work and throughout my life. My beloved parents also have given me their unequivocal support throughout, as always, for which my mere expression of thanks likewise does not suffice. Their prayers, guidance, and advice have been valuable throughout my life. This thesis is a significant achievement that belongs to them. My special gratitude also goes to my brothers and sisters.

Finally, to other people who are involved directly and indirectly with this thesis, I would like to state my appreciation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Motivation

Reconfigurable hardware or ReHW has become an important research field in electronic systems due to the advantages of rapid prototyping [1] and flexibility [2]. ReHW allows reduction of the development cost, efficient resource sharing, and hardware reuse. In a power-aware system, scalable and dynamic power management can be implemented, leading to eco-friendly systems. A reconfigurable architecture ensures that hardware can be switched easily between several operational modes. Its flexibility to adapt to different modes enable a system to reduce its energy consumption, and hence its power dissipation, when it is configured to operate in stand-by mode in comparison with full operation mode. This feature is also important for green-device technology, as unnecessary heat emission to the environment can be reduced.

In wireless applications, especially battery-operated devices, such advantages are important to improve the dynamic performance of a system and prolong its power supply life. In the future, when multimedia communication dominates wireless applications with many functionalities and operation modes, reconfigurable hardware will become essential.

In modern digital devices, the analog-to-digital converter (ADC) is an important module. It transforms time-varying analog signals into a digital format for signal processing, data enhancement, and storage. Since the ADC is crucial for modern electronics, a consistent research effort has been aimed at improving the performance of ADCs using the reconfiguration approach. The signal-to-noise ratio (SNR) can be increased proportionally with the resolution of the ADC. Any extra resolution will improve the SNR by approximately 6 dB [3,4]. On the other hand, power consumption of an ADC increases with the increment of resolution N, and sampling rate f_s [5]. In the future, when wireless communication supports multimode signalling and multiple applications, adaptively changing these ADC parameters with a single ADC architecture may be useful for many applications. Furthermore, reducing N and f_s during idle states could reduce the power consumption of an ADC. The power consumption issue regarding an ADC is significant for battery-operated portable wireless devices. It is vital to limit the energy use to prolong the battery's lifespan. Reducing the power consumption of ADC also minimises the heat dissipation from the device, and simultaneously reduces the energy emission to the surroundings. The operation of internal circuits also determines the power usage of ADC. In principle, if a certain block can be disabled when not required, its dynamic power consumption will be reduced. Such an implementation method is only applicable with the use of reconfigurable technology.

While designing reconfigurable ADCs (ReADCs), several challenges must be overcome. One important reconfigurability issue is the configuration headroom introduced by the reconfiguration mechanism. Extra modules are required as

configurable switches, a controller, redundancy modules, or a selection of different architectures for various operations. Another issue to address is detection-and-respond mechanisms, specifically important for real-time or adaptive reconfiguration. Additional area and power are necessary for reconfigurable architecture to implement the control mechanism that monitors and responds to the specific requirements. This is a critical issue for real-time reconfigurable hardware, as dynamic performance is related directly to this mechanism and affects its overall performance efficiency significantly.

Current research works in ReADCs are focused mainly on different architectural types and the analog circuit techniques. The adaptive mechanism is assumed to be triggered by baseband processing parts that send a configuration signal to the ReHW. Detailed discussion on adaptive feedback controller mechanisms and digital reconfigurable control schemes is still limited. This thesis proposes a novel adaptive control scheme for ReADC targeting low-power portable devices. It focuses on adaptive control schemes at both algorithmic and architectural levels. Particular attention is given to adapting the ReADC operation according to the variations of the signal amplitude.

1.2 Thesis Objectives

The objectives of this thesis are to investigate the following two important reconfigurability issues of ReADC for universal mobile telecommunication system (UMTS) applications: i) the detection mechanism that supports real-time reconfiguration, and ii) the area and power overheads incurred by circuitry due to configurability of the devices. Consequently, low-power solutions for real-time ReADC are suggested.

To analyse the above two issues, the following eight steps are undertaken:

- To determine the minimum requirements of ADC parameters, according to application specifications.

- To study the UMTS specifications, and to generate and analyse the UMTS signalling.

- To develop an adaptive algorithm for ReADC according to the ADC digital output.

- To demonstrate the capability of reconfigurable ADC to change its parameters according to signal variations.

- To analyse the performance of the adaptive algorithm on different parameter settings.

- To develop low-power adaptive control hardware for ReADCs.

- To investigate the effect of the proposed method in ADC-based systems for noise correlation and detection.

- To improve the efficiency of the data acquisition method using reconfigurable technology.

1.3 Thesis Structure

The rest of this thesis is organised as follows:

• Chapter 2 provides background information of the thesis. It covers the review of reconfigurable hardware technology. It is followed by an overview of spread spectrum communication. Specifically, wireless transceiver architecture for UMTS applications is highlighted. Next, typical ADC topologies, which are used in such battery-operated wireless systems, are presented. An aspect of low-power design is discussed before the power evaluation of ADC is presented.

- Chapter 3 discusses the existing reconfigurable ADCs, including the ones used in wireless communication and UMTS applications. Current trends in adaptive control mechanisms are also presented.

- Chapter 4 describes the specifications of the ADC for UMTS applications. In particular, minimum resolution N_{min} and minimum sampling-rate $f_s(min)$ values are derived. Both parameters determine the lowest possible N and f_s values of the ADC selection for such applications.

- Chapter 5 presents a technique to achieve real-time reconfigurability for a ReADC, specifically for UMTS applications. Initially, the UMTS signal is generated and analysed in terms of its periodicity and framing structure. Then, the ADC behaviour and its output mechanism are observed. Consequently, an adaptive algorithm is proposed in this chapter. The algorithm that is based on switching activity SA computation of the output of the ADC is known as a switching activity measurement (SWAM) algorithm. The justification of this algorithm is analysed and simulation results showing the effectiveness of this technique are discussed in this chapter.

- Chapter 6 analyses the impact of varying parameter settings on the proposed adaptive algorithm. Particularly, the response of the algorithm is evaluated according to an observed interval and threshold value settings. The effect of changing the value of the spreading factor (SF) at the baseband on the bit-to-error rate (BER) performance is also monitored.

- Chapter 7 presents the hardware implementation of the adaptive algorithm. Initially, power estimation models for ReADCs are described as the performance metrics. The hardware performance of the suggested application-specific integrated circuit (ASIC) module for adaptive N is also presented.

- Chapter 8 describes other potential uses of the proposed SWAM method to detect different levels of noise strength in ADC-based electronic systems. In addition, the effect of weighted coefficients on the algorithm are also analysed and

a low-power noise detection module is suggested.

- Chapter 9 summarises and concludes the thesis. In addition, the contributions of the work are re-highlighted and further research possibilities based on the techniques developed in this thesis are evaluated.

A summary of the chapters and their interconnection is presented in Fig. 1.1.

1.4 Summary

Multimode operations and mixed-signal applications are important for future wireless communication. In this case, different kinds of wireless signal can be sent over a single transceiver module. Hence, this module must be configured to suit different signal variations, operation modes, conditions, and applications. This thesis proposes an adaptive method for ReADCs targeting low-power portable devices. Furthermore, the study focuses on developing an adaptive control scheme at the algorithmic and architectural levels. Particular attention is given to reconfigurability issues and the interactions between the ReADC architecture and its adaptive control mechanism. Moreover, this thesis also investigates the adaptive performance of the ReADC according to time-varying signal variations. Finally, the proposed SWAM method is investigated for other ADC-based systems targeting low-power applications and power-aware green device technology.

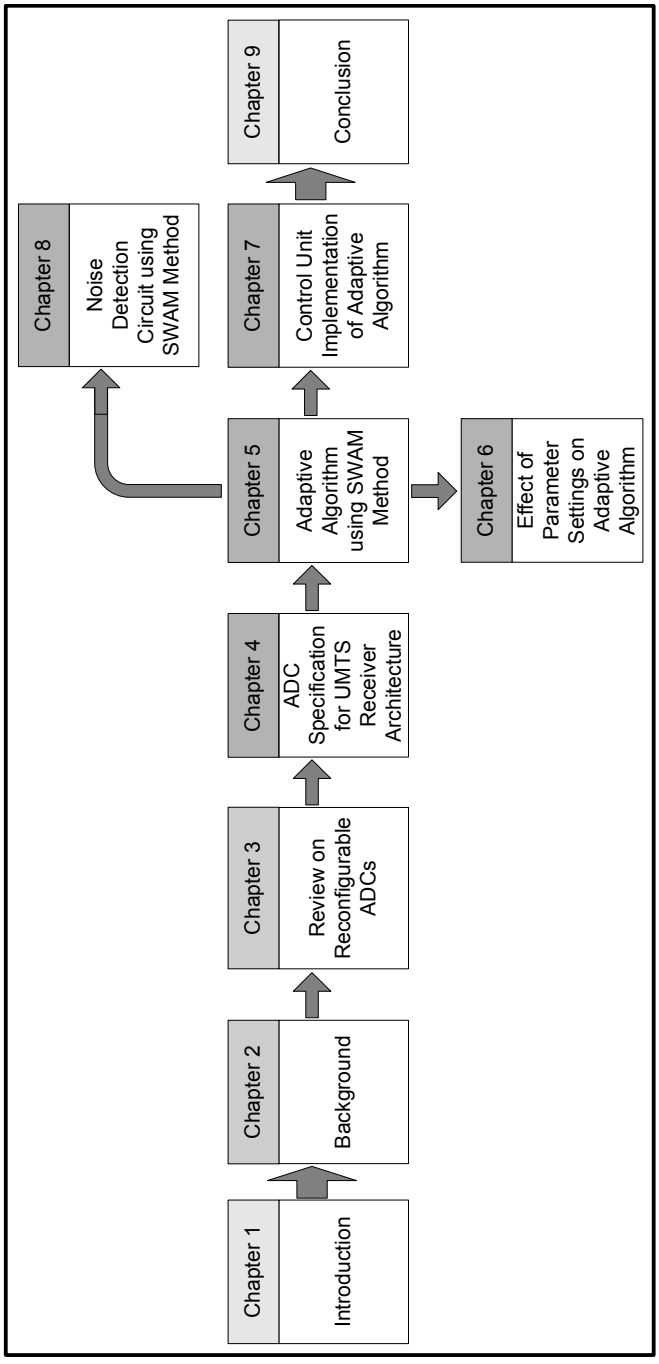


Figure 1.1: Flow of the thesis

Chapter 2

Background

Contents

2.1 Introduction

This chapter presents the relevant background information for this thesis. In Section 2.2, a general overview will be given regarding the importance of reconfigurable technology and ADC choice in future wireless communications. Section 2.3 will discuss the emergence of ReHW as a branch of electronics system design. The motivation behind this technology will then be reviewed to see how it affects

current applications. Real-time ReHW will be evaluated, followed by the challenges arising from the design. Section 2.4 will provide an overview of wireless technology. It starts with the review of the modern mobile communication system, including transceiver architecture for portable devices. Suitable digital converters for the architecture will also be discussed. In Section 2.5, a review on ADCs, including the choice of ADC for specific applications, will be presented. General subcircuits for ADC will also be described, including a brief discussion on pipeline ADC targeting wireless applications. Section 2.6 will describe low-power design from the perspective of complementary metal oxide semiconductor (CMOS) process technology and circuit design approach. Several design techniques have been implemented to continuously reduce the power consumption of electronic circuits. In Section 2.7, the power evaluation of ADC at system and circuit levels will be discussed. This provides an estimation of the power consumption of ADCs depending on N and f_s values, and other design parameters. Section 2.8 describes the design flow of the works. Finally, Section 2.9 summaries this chapter.

2.2 Overview

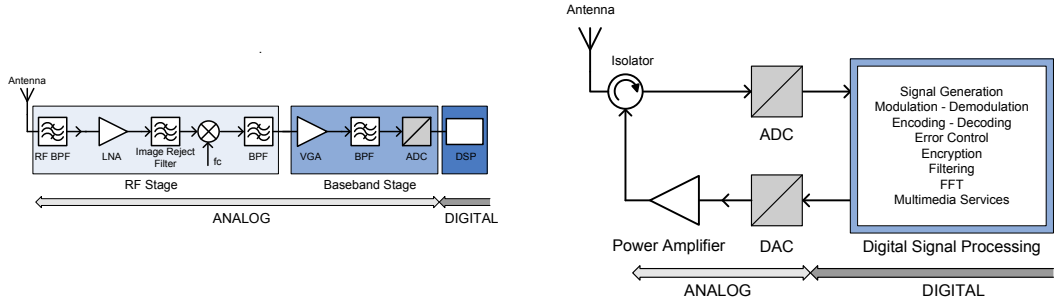
Cellular communication has experienced rapid progress in the last decade. This is due to the main objective of today's wireless communication: to provide an excellent quality of service (QoS) at user terminals. This has been the target of design trends for wireless technology, especially towards a higher data rate with the integration of multimedia services. In hardware implementation, the receiver is designed as a high performance system with small footprint topology for low-power consumption and high noise immunity. Aside from that, the integration of digital signal processing with analog front-end modules has also influenced these improvements. From a circuit design perspective, it is desirable to push the digital domain closer to the antenna to benefit more from digital processing. In this case, both transceiver architecture and targeted wireless applications influence

the requirement for an ADC. In Europe, UMTS has been implemented for third generation (3G) wireless communication. It uses wideband CDMA (WCDMA) technology as its driving engine to increase the data bandwidth for high-speed multimedia applications. UMTS specifications are provided by a governing body called the 3rd Generation Partnership Project (3GPP) as a standard reference for the design engineers.

Meanwhile, advances in modern digital communication systems have contributed to significant improvements in signal quality over analog systems. Digital signal processing, such as the fast Fourier transform (FFT), coding, encryption, error detection and correction, together enable better signal immunity, protection, and analysis; this would not be possible or difficult in analog communication. Furthermore, modulation and filtering are much easier in a digital than in an analog domain. However, due to the analog nature of real-world communication, the use of an ADC in electronic devices is essential. The ADC is an interface between the analog front-end and digital back-end circuits. In digital wireless communication, the analog front-end is used to shift the signal frequency to suit channel capacity. It involves processing of small analog signals that involves amplifications using low-noise amplifier (LNA) and variable gain amplifier (VGA), filtering using band-pass filter (BPF) and image-reject filter, frequency shifting using carrier frequency f_c and mixer, and signal conversion using ADC. Digital back-end or the baseband stage involves digital processing and signal recovery. Figure 2.1(a) illustrates the typical wireless receiver architecture.

Since digital signal processing is more powerful than analog processing, current research trends in wireless technology are pushing the analog chain closer to the antenna, as shown in Fig. 2.1(b). Essentially, the filtering and frequency conversion of the analog front-end are implemented in the digital domain after ADC. Such an approach leads to the software-defined radio (SDR) concept [6]. With multimode applications, a wider bandwidth is required to accommodate higher data capacity. Thus, the main design challenge for ADC design for future wireless

communication is to achieve a high dynamic range (DR) requirement. Furthermore, since the received frequency range in SDR is wide, disturbance and noise issues become significant.



(a) Typical wireless receiver architecture (b) Ideal software radio receiver

Figure 2.1: Wireless receiver architecture

2.3 Reconfigurable Technology

During recent years, reconfigurability has been an important paradigm in electronic system design. Due to the extensive growth in portable devices and wireless communication in supporting multiple applications and multi-mode operations, the design tasks have become more complex. Similarly, in physical implementation, the chip area grows larger; therefore, more power is required from the supply. These design efforts are not only to minimise the chip area for portable devices, but also to limit the power consumption. This ensures that the devices can operate for much longer periods of time. To overcome these constraints, re-use design methodology is seen as a way forward. Reconfiguration has not only emerged as an efficient solution for dynamic power management in multi-standard applications, but it also plays an important role as a resource-sharing mechanism. A reconfigurable system also enables particular architecture to change its operation behaviour to meet the different kind of requirements. During the early stage of design, reconfigurability helps to speed up the time to market by allowing rapid

prototyping of a product.

With the increasing popularity of wireless portable devices, and as more applications are integrated into a system, the circuit footprint enlarges and power consumption increases dramatically. Hence, ReHW will not only help to reduce the chip area by using the resource-sharing concept, but it also has the advantages of dynamic performance optimisation and energy-efficient usage. To implement the reconfiguration concept successfully, additional reconfigurability issues must be considered at the different design layers.

2.3.1 Overview of Reconfigurable Hardware (ReHW)

At the moment, the evolution and active development of electronics show no signs of slowing. This is mainly due to increasing demands from modern consumer electronics pushing the design boundaries continuously. Motivated to increase the quality of life, electronic design issues are often centred on speed, power, performance, and robustness. With the increased popularity of portable devices, electronic appliances are required to operate with minimum power while keeping adequate performance and speed. These design parameters are usually a trade-off between power, performance, and speed requirements. The traditional design approach makes it difficult to optimise the solution for all parameters. In particular, once a design has been realised in hardware, then it is impossible to change the design functionality. To increase the flexibility of a design, a reconfigurable approach has been introduced. Initially, the reconfigurable method was implemented in digital design to speed up implementation and reduce development cost. The reusability of the hardware is also beneficial to optimise hardware usage in different data processing modes. Influenced by this flexibility, reconfigurable techniques have also been implemented in analog design during the last decade. In general, ReHW is used to improve five key areas of the design: 1) *flexibility*; 2) *performance*; 3) *power consumption*; 4) *non-recurring engineering (NRE) costs*; and 5) *programmability* [7].

2.3.1.1 Evolution in Hardware Design

Electronic design influences our daily lives tremendously. Since the invention of the junction transistor in 1947, development in electronics has evolved constantly, and continues today. Similarly, the evolution and revolution of electronics have created many design opportunities and new research areas.

The demand to improve the quality of daily lives has created research opportunities in many aspects, including electronic design. Software, hardware, analog, digital and mixed-signal designs have all evolved rapidly. The main research issues consist of cost reduction, time-to-market, performance, operation speed, chip area, size, power utilisation, robustness, and flexibility. Designers work continuously to improve all these aspects. Hence, many design methods and circuit approaches are being developed. Usually, there are trade-offs between design issues and therefore, one aspect has to be sacrificed for the other. For example, a high performance device consumes high power to operate.

Since mobile phones entered the market in the 1980s, interest in portable devices and consumer electronics has grown tremendously. This scenario drives research even further as the design issues become more important. However, the main goal is to improve the design without having to compensate too much within the design parameters. Portable devices require smaller chip areas and consume lower power but must simultaneously maintain performance and speed.

In the last decade, reconfigurable analog design was introduced to achieve greater design flexibility [8, 9]. Originally, the reconfigurable digital approach, such as field programmable gate arrays (FPGA) and reconfigurable computing (RC) from the late 1980s inspired its development [10,11]. Hardware reconfiguration also enables design robustness [12] and post-production programmability [11]. For example, genetic programming (GP) [13], field programmable analog arrays (FPAA) [9, 14], and field programmable transistor arrays (FPTA) [8, 15] use a genetic algorithm (GA) concept to evolve to suit the changing environment via system learning and to improve performance over time [16]. This also helps the cir-

cuit to operate in challenging environments, such as remote areas and unmanned space operation [12].

2.3.1.2 ReHW Concept

The emergence of FPGAs in the mid 1980s led to an era of ReHW [10]. Since then, it has become one of the most active research areas for various applications. ReHW enables its architecture to be programmed after fabrication to perform different kinds of operations. Hence, a single architecture can execute multiple operations.

ReHW has been used to increase the flexibility of an electronic system to suit to different application modes, environment conditions, and constraints [17]. An adaptive system is capable of monitoring the changes and adjusting its operation dynamically according to specific references. Such a system not only reduces its internal circuit activity, which leads to reduction in power consumption, but also manages the utilisation of hardware resources efficiently. In a challenging environment, adaptive hardware configuration is also able to overcome excessive interference signals and suppress unwanted signals in noisy surroundings.

2.3.2 Advantages

ReHW has attracted much research interest due to its many advantages. Firstly, it allows the operation of hardware to be flexible. The architecture is not fixed to a particular operation; it can be used for different operations depending on the requirements, by switching to a different architecture configuration. Secondly, ReHW also supports reusability mechanisms. An element in ReHW is activated in an operation only when necessary. Configurable elements (CEs) can be set to idle mode when not in use.

A real-time reconfigurable or adaptive mechanism is another major advantage of ReHW. Adaptability is achieved by dynamic monitoring of a continuously changing environment. Additionally, a control mechanism is required to evaluate

responses to the dynamic changes.

The efficiency of an operation can also be increased depending upon its application. In these cases, the time to perform a task can be changed according to the complexity of an algorithm. Another aspect of reconfiguration is cell granularity, in which power consumption can be scaled to a minimum usage. With ReHW, speed and power trade-off can also be optimised.

The performance of ReHW can be improved when necessary. In remote areas and unmanned operations, reconfiguration becomes important to increase the robustness and to ensure a fault-tolerant device. At the early stage of the design, reconfigurability aids the engineers to speed up the time-to-market and enable rapid prototyping.

2.3.3 Configuration Approaches

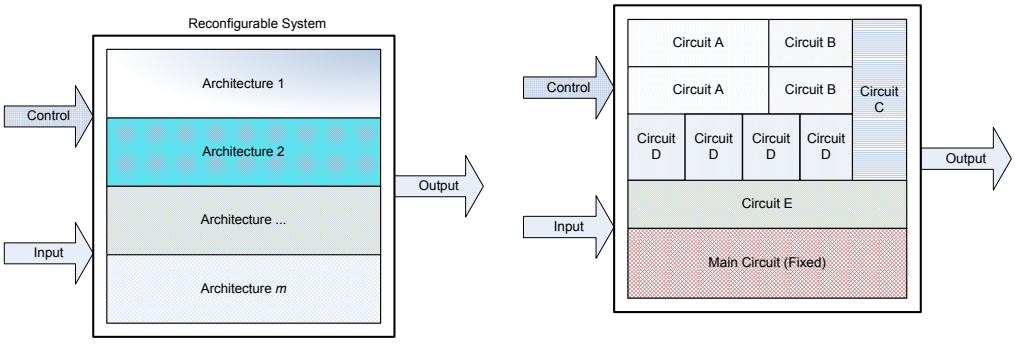
Configuration technique is an important aspect in ReHW. The architecture of ReHW consists of different CEs depending on the different reconfiguration layers. This architecture relies on external control signals as configuration information.

2.3.3.1 Level of configuration

The configuration can be implemented at different abstraction levels, such as algorithmic (system), architecture, circuit, or parameter layer, as each layer has a different complexity.

Instead of an arrangement of a standard granular CE of the same functionality, ReHW can also consist of sets of pre-defined architectures, as shown in Fig. 2.2(a). Normally, each architecture performs a different kind of function. The external control signal determines the selection. In some cases, the architectures can be in different topologies, but offer similar functionality. These hardware redundancies can be used as back-ups in error-prone conditions, or are used to adapt to different constraints, particularly in unmanned operations and remote area applications.

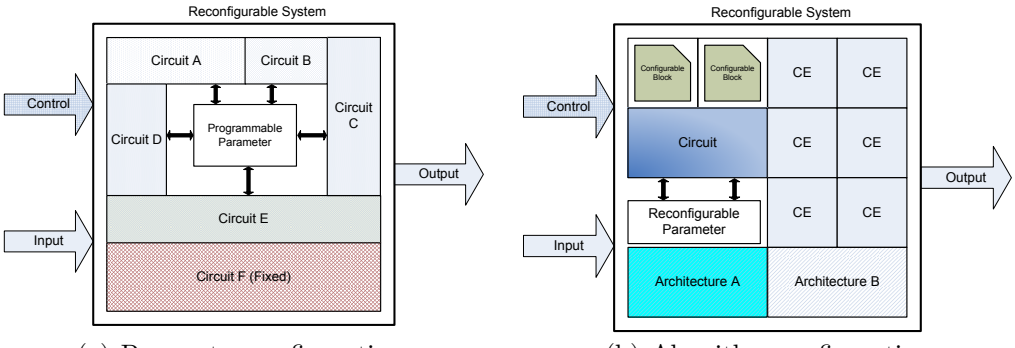
In the analog domain, ReHW normally consists of configurable circuits (Fig.



(a) Architectural configuration (b) Circuit configuration

Figure 2.2: ReHW with architectural and circuit configurations

2.2(b)) or tuneable circuits (Fig. 2.3(a)). Architecture with configurable circuits programs the pre-defined modules, such as amplifiers and current sources, to perform specific functions. Meanwhile, architecture with programmable circuits normally tunes parameter values. An example of this approach is changing capacitors' or current sources' values. With different combinations of architecture, circuit, and parameter reconfiguration (Fig. 2.3(b)), algorithms at the system level can be tuned to optimise the operation.



(a) Parameter configuration (b) Algorithm configuration

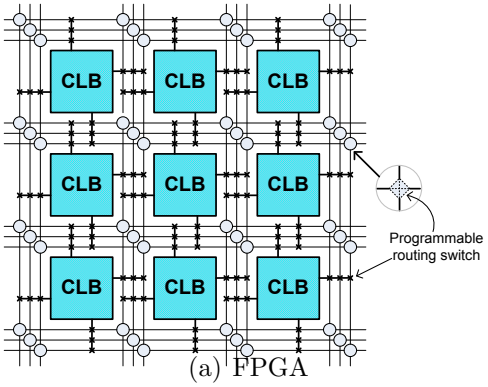
Figure 2.3: ReHW at parametric and algorithmic configurations

2.3.3.2 Methods

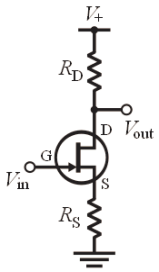
A common approach to the ReHW utilises configuration switches [18]. An example of this approach is in FPGAs, when the clustered logic blocks (CLBs) are

configured by using a switches matrix, as shown in Fig. 2.4(a). In this case, control signals contain information regarding which switches need to be activated. Normally, this approach is implemented in the digital domain, where control signals contain binary bit strings of programmable information.

Another approach is to set a reference value and allow the circuit to configure the device automatically. An example of this approach is implemented in the analog domain. As seen in Fig. 2.4(b), the source resistor R_S sets the source voltage of the CMOS transistor. If the current flowing through the channel I_C is increased by δI_C, then the gate-source voltage V_{GS} is also increased by δV_{GS}. However, the voltage drop around R_S also increases, pushing the source voltage V_S higher. As a result, V_{GS} is decreased gradually until the current reaches a stable state.



(a) FPGA



(b) Common-source amplifier with source degeneration

Figure 2.4: Examples of configurable approaches

2.3.4 Application of ReHW System Design

With its benefits, ReHW has become the backbone for a wide range of applications, such as software-defined radio, medical imaging, networking, encryption, scientific data acquisition, spacecraft, robotics, automotive, imaging, and multimedia [17]. In these applications, ReHW has improved solutions on time-to-market, design cost, fault-tolerant, and computation. Similarly, certain issues on

performance, power-and-speed trade-off, resource sharing, adaptability, area footprint, and remote-area operation have been resolved using this reconfiguration technology.

Applications of ReHW, such as RC [19], use the reconfigurability technique to speed up its complex processing with parallelism. Meanwhile, reconfigurable digital circuits such as fault-tolerant devices [20, 21], help tackle problems due to cosmic rays from outer space. Furthermore, reconfigurable analog circuits like ReADCs are used in wireless communication [22] and wireless sensor networks [23]. Future mobile communication frameworks, such as SDR, have also been constructed with reconfigurable technology [24, 25].

2.3.5 Adaptive Hardware System

ReHW can be either configured off-the-shelf (*offline*) or on-the-fly (*online*) [26]. When ReHW is configured offline, the reconfigurable instruction is loaded to program the architecture before it is implemented in a system. When a different function is desired, a new configuration instruction is loaded while the system is not operating to avoid any interruption. However, this kind of reconfigurable mechanism does not respond to the sudden change of interference or environment.

To allow a reconfigurable system to react with its surroundings, a real-time or adaptive mechanism is used. This online reconfiguration changes its operation according to sets of rules, references, specifications, or predefined functions. The system is also capable of learning and observing changes by comparing the results of the current state with the previous state or the stored references. Therefore, this dynamic approach may be defined as a 'learning system'.

However, real-time ReHW or adaptive hardware (AHW) has to deal with certain reconfigurability issues, which are less critical in the *offline* ReHW implementation. For AHW, an additional circuit area is necessary to accommodate references and system observations. Similarly, adaptability requires headroom in terms of power and area for controlling the programmable devices.

The architectural difference in AHW is the requirement of an additional controller unit as a part of the system configuration, apart from the normal ReHW architecture (Fig. 2.5). This control unit is used to sense or detect, evaluate, and decide on the ReHW configuration [17]. In detail, a typical reconfigurable architecture consists of the main CEs, memory elements (MEs), and a control unit. It also includes buffers and interconnects as part of the reconfigurable processing unit, as described in [26].

In some applications, the monitoring process is operated at the software level, in which the controller unit is implemented on a PC. Since implementing the controller at the software level takes a longer data path than in hardware, the response of such an application is normally slower. Moreover, the power introduced by the controller part is higher than the reconfigurable device itself, by assuming the device is implemented in ASIC or is integrated in a smaller board.

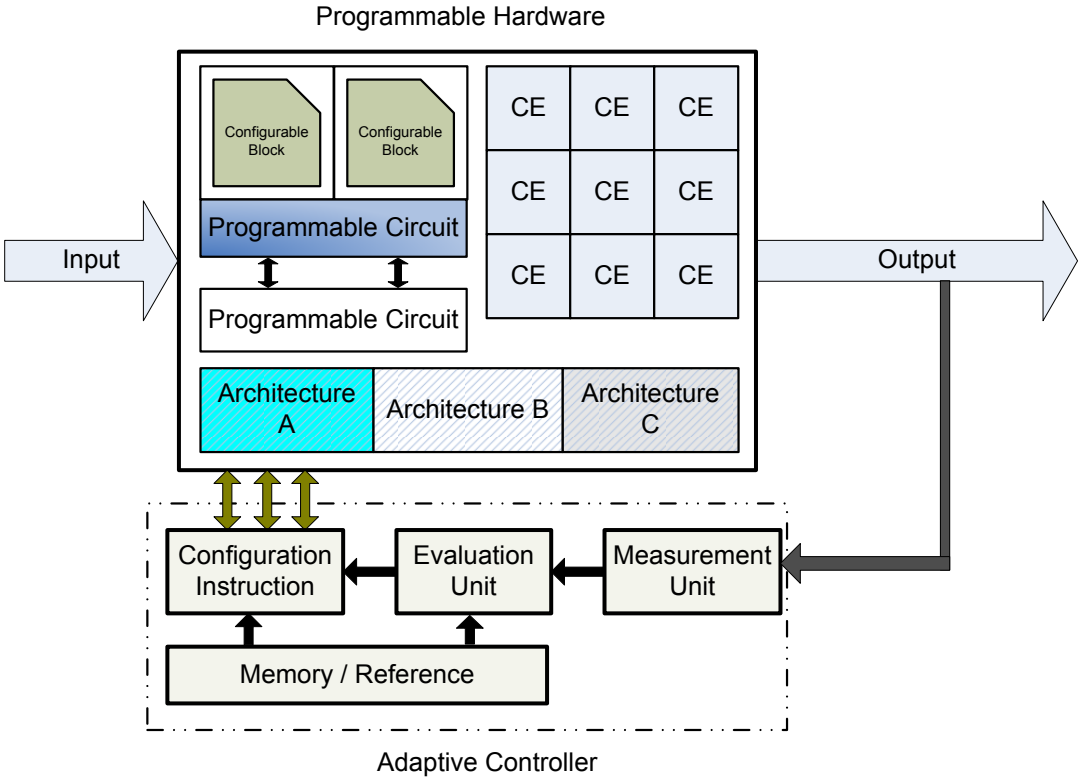
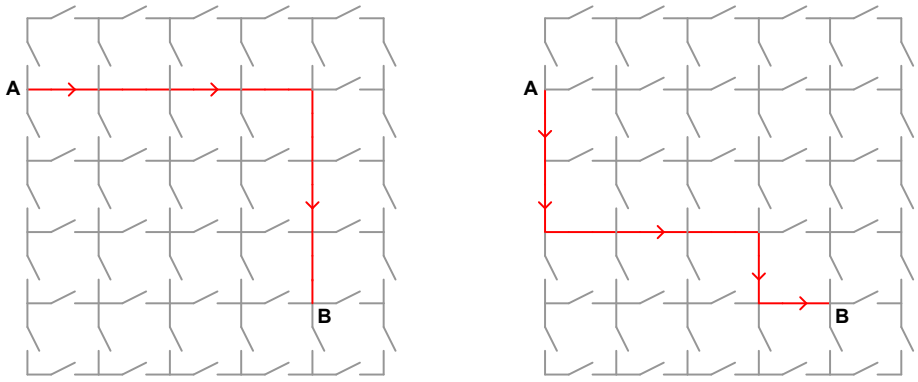


Figure 2.5: Architecture of adaptive hardware

AHW uses the feedback concept to attain the required functionality. A switching circuit is used, where the control information contains switching data for reconfiguration. Similar to the ReHW methodology discussed in Section 2.3.3, AHW uses configuration or selection switches to re-program, re-use, and re-route the system, as shown with an example of configuring from '**A**' to '**B**' in Fig. 2.6.

Firstly, the controller measures the output signal. Then, the data is evaluated to obtain specific information regarding the configuration. The output of the process is compared with reference values or functions stored in memory cells. As a result, the controller sends control signals to the reconfigurable architecture.

The control signals contain information to trigger the switching arrays to 'ON' and 'OFF'. While in digital applications, configuring the switches does not affect the function; in analog applications, the switches should not leak any current due to switching to avoid system malfunctions.



(a) Switching route 1 (b) Switching route 2

Figure 2.6: Examples of switching and re-routing

2.3.6 Design Challenges and Reconfigurability Issues

To implement AHW successfully in a system, designers face reconfigurability challenges at different design stages. Choice of application, design process and tools, configuration methods, and system implementation all have different issues that need to be considered. Typical hardware design issues are performance trade-offs,

circuit immunity either from noises or faults, reliability of the design, design for manufacturability, chip form factor and area, and implementation cost.

Although ReHW has been implemented in a wide area of applications, various design issues must be taken into consideration before the hardware is developed. While ReHW has advantages in flexibility, reusability, and performance improvement, the issue of overhead may limit the realisation of this technology. Furthermore, in an adaptive implementation, circuit response is critical since it determines the adaptability of the hardware to the dynamically changing environment. Similarly, interoperability of wireless standards has become one of the most important challenges in the design of current and future reconfigurable architectures for wireless systems [27].

In summary, ReHW is important for electronic system designs due to its many advantages. With portable devices currently dominating the market, the requirement to stretch trade-offs between speed, power, performance, reusability, flexibility, granularity, and computation efficiency becomes crucial. The reconfigurable approach is not only helpful during the design stage, but may help to solve many post-production problems and enhance the dynamic behaviour of a system. Undoubtedly, ReHW is a strong contender to solve multiple applications and multi-mode operations problems for portable devices. However, designing a reconfigurable system introduces additional design issues that should be examined carefully and rigorously before such a system can be implemented successfully.

2.4 Wireless Technology

According to the International Telecommunication Union's (ITU) statistic released in 2010 and depicted in Fig. 2.7, the number of mobile users has grown drastically in the last decade [28]. From just under one billion at the beginning of 2000, the mobile phone market soared to exceed five billion users in 2010. Similarly, the penetration of mobile users per 100 inhabitants has also doubled

every four years for the last eight years. This increasing demand requires room for improvements in terms of costs (i.e., development, production, maintenance, affordability), QoS, and research and development (RND).

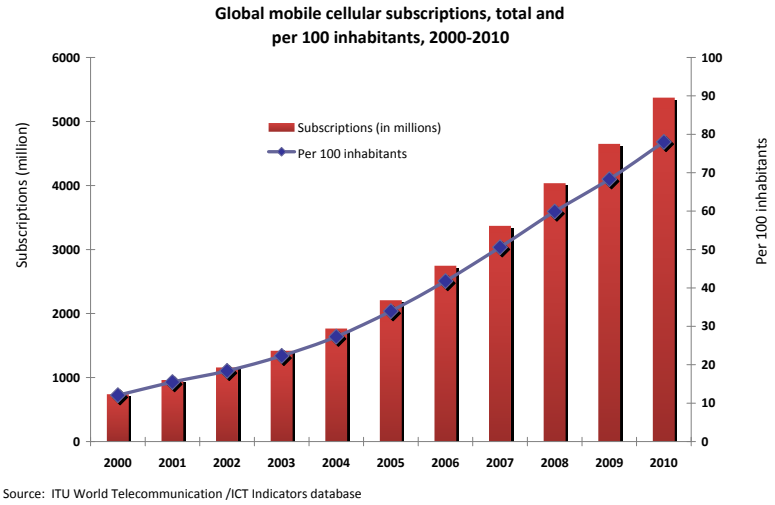


Figure 2.7: Global penetration of mobile users

Since the invention of the first mobile phone, the world of wireless communication has developed very fast, with a bright future predicted. For each generation, new technology has been adapted to improve its performance. Starting with a full analog operation, followed by enhancement with digital operation, wireless products link people together and close the gaps.

Wireless protocols have also experienced major rework. UMTS, the release of 3G system standard, is implemented to encounter problems from the previous generation. A new wireless technique has been adapted to this system. Code division multiple access (CDMA), a currently popular multiple-access type, is selected to be the core function of the UMTS . It has the advantage of higher user density inside a cell compared with the previous wireless generations. It has also gained attention due to its low-power consumption, leading to many advantages.

UMTS is one of the five members of the international mobile telecommunication 2000 (IMT-2000) family of standards, which relies on the use of spread

spectrum technology as its main operational core. Focusing on better QoS, UMTS has many advantages over the well-known global system for mobile communications (GSM) and enhanced data rates for GSM evolution (EDGE) technologies in terms of power consumption, user density per cell, maximum data transfer, very low probability of interrupts, and excellent channel diversity [29, 30]. It also has the potential for adaptation to other new standards worldwide.

There are many choices for the transceiver front-end. Despite the fact that many different transceiver modules can be used, the choice should be simple and easy to implement. The selection of the transceiver architecture for UMTS can also influence the overall performance. From the wide variety of choices, the design requirements of low-power consumption and chip size have become the main constraints. Radio frequency (RF) front-end designer choices are usually towards a simple architecture for easy implementation; yet, it retains its performance. Furthermore, since the complete transceiver consists of several building blocks, any sensitive block should be identified and isolated from other noisy parts. This will ensure the correct operation of the system.

2.4.1 Universal Mobile Telecommunication System (UMTS)

UMTS is developed based on a global multimedia mobility (GMM) system architecture family, which supports flexible and integrated high-speed access for mobile communication [31]. This 3G telecommunication system is based on WCDMA as the radio interface [30], as WCDMA is characterised by the use of a wider bandwidth than the normal CDMA. Its spectrum is 3.84 MHz wide and supports two signal components: in-phase I and quadrature Q with chip rate R_c of 1.92 Megachips per second (Mcps) each.

2.4.2 UMTS Frequency Band and Operation

The UMTS frequency division duplex (UMTS/FDD) mode is one of five standards within the IMT-2000 family concept adopted and agreed upon at the end of 1990

Table 2.1: Frequency assignments for UMTS

	Frequency Allocation (MHz)
Uplink Frequency	$1920 - 1980$
Downlink Frequency	$2110 - 2170$
Tx – Rx FDD Separation	190
Rx / Tx Channel Width	60
Rx / Tx Channel Spacing	5

[32]. The other four family members are categorised as multi-carrier (MC), time-code (TC), single-carrier (SC), and frequency-time (FT). UMTS is driven by direct sequence CDMA (DS-CDMA). Orthogonal codes [33] are used in DS-CDMA, and the orthogonal Walsh code [34] is the most widely implemented.

In Europe, IMT-2000 allocates frequencies of 1900 to 1980 MHz for the UMTS transmitting channel and 2110 MHz to 2170 MHz for the receiving channel with 190 MHz for UMTS/FDD channel spacing [30]. A transmitting channel bandwidth of 60 MHz is chosen, starting from 1920 MHz. Although the upconverted data bandwidth is actually 3.84 MHz, 5 MHz channel spacing is reserved for the receiver (Rx) and transmitter (Tx) bandwidth, giving an extra 23.2% for the guard band. Therefore, the total 60 MHz bandwidth for the transmitter and receiver results in 12 selective channels. Table 2.1 presents a summary of the frequency allocation.

The channel spacing of 5 MHz contains the chip rate R_c of 3.84 Mcps bandwidth. This rate is modulated with quadrature phase shift keying (QPSK) and is filtered with a root-raised cosine (RRC) filter with a roll-off factor of 0.22. It carries approximately 99% of the total power of the channel.

The RRC channel filter defines the shape of the transmitted data. The channel is tuned to the specific frequency by using frequency synthesiser consisting of a phase lock loop (PLL). The locked signal is then downconverted at the RF front-end and demodulated at the back-end (reverse processes for transmitting signal). This analog signal is transformed subsequently into the digital domain for data

processing.

2.4.3 Zero-IF Receiver for UMTS Application

Recently, Zero-IF or a direct-conversion transceiver has become the focus of UMTS applications. Even though its principle and architecture has been known for a long time, the problem of DC offset and local oscillator (LO) leakage has prevented its realisation earlier than expected. The principle of Zero-IF, which is also known as *homodyne* receiver, is depicted in Fig 2.8. The wanted signal at the antenna is converted directly to the audio frequency using a local oscillator, without the need for an intermediate frequency (IF). Therefore, high DC components heavily distort the signal. This is one of the main challenges for Zero-IF implementation. Other problems are I-Q mismatch, channel selection, even-order distortion, and flicker noise $^1/_f$. Only recently, with numerous researches focusing on these aspects, has this architecture been successfully realised.

Despite of all these drawbacks, the advantages of Zero-IF receiver are more significant. It is the obvious choice compared with other architectures, as its simple topology requires less functional blocks. The bulky and expensive surface acoustic wave (SAW) and image filters are rendered unnecessary. From the mobile communication perspective, Zero-IF is perhaps the best architecture in terms of being compact and small. This is suitable for single RF chip solutions since it allows better monolithic integration. Therefore, this advantage is the leading factor for lightweight, high performance, and mass volume production of RF front-end chips for mobile phones.

A popular topology for the UMTS/FDD mobile station (MS), or 'mobile phone', is the Zero-IF architecture, since it requires less analog blocks for the down-conversion [35]. Its small footprint indicates that less circuit area is occupied, which is suitable for portable devices. Moreover, a small circuit area may lead to lower power consumption.

A user channel bandwidth of 3.84 MHz is split into two components: I and

Q signals, respectively. Both I and Q carry 1.92 Mcps in 2.5 MHz bandwidth, half the single channel bandwidth. The I-channel consists of data bits while the Q-channel contains modulated and spread control bits [36].

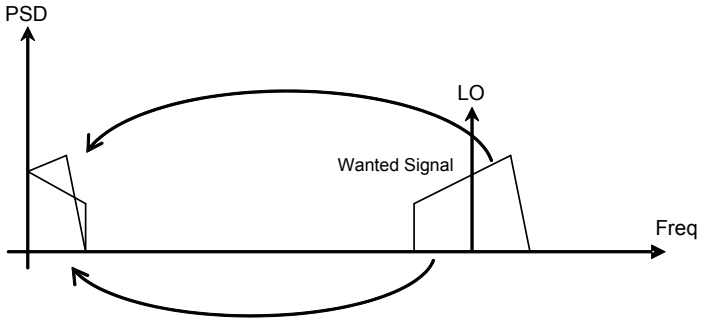


Figure 2.8: Principle of the Zero-IF receiver

Figure 2.9 illustrates the topology of the Zero-IF receiver. A high frequency analog bandpass filter is used to select a specific bandwidth and reject other frequencies. An LNA amplifies the received UMTS signals. The amplified signal is then coupled into two downconversion paths. One part is multiplied with cosine signal and the other with sine signal. The conversion result then contains similar frequency components as the original signal. Channel filters are included to block any unwanted components other than the desired channel. A voltage gain amplifier (VGA) is used to adjust the signal amplitude suitable for the digitisation process by the ADC.

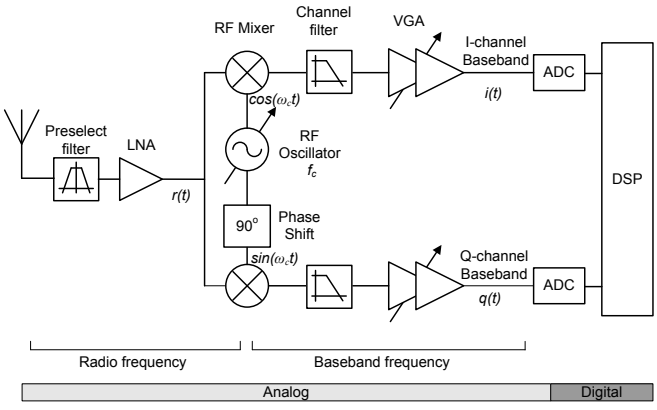


Figure 2.9: I-Q direct downconversion (Zero-*IF*) receiver topology

2.5 Analog-to-Digital Converter (ADC)

The digital domain enjoys the benefits of digital signal processing, including data protection, encryption, enhancement, storage, and management that can be performed with ease. Therefore, there is a need to convert analog signals into the digital domain to use the advantages of the digital signal processing.

An ADC is a very important module in the modern electronics device [5]. In fact, it is unavoidable in the current digital era. It is the heart of digital signal processing, as the manipulation of the desired signals is much easier and powerful than in the analog domain. The ADC, the interface between the two domains, takes an analog sample and converts it into a string of the digital output. This output is then fed through into the digital circuitry for further processing. The length of this output string for each sample, also called ADC's wordlength, is dependent on the ADC resolution, N. As N increases, the precision of the analog amplitude represented by a digital bit string will be closer to the original source.

In general, the ADC architecture composes of three stages: sampling, quantisation, and coding, as shown in Fig. 2.10 [37]. In the sampling process, the incoming analog signal is sampled (i.e., analog amplitude is read periodically) at a fixed time interval. Normally, sampling is done together with a 'hold' circuit. This part ensures the digitisation process of the current sample is completed before the next sample reading. The sampled signal is then quantised in a ladder step. The ladder can either be linear or nonlinear steps. Each step will correspond to specific digital code. Since there is a variation between continuous analog signal and quantised ladder, an error exists, known as the quantisation error Q_ε, and it needs to be as small as possible. As N increases, the ladder steps will also increase. Consequently, the accuracy will improve with the ladder steps, which represents N. The final stage of digitisation is the coding, which is the mapping of the sampled analog signal amplitude to the digital word corresponding to each ladder step. The analogue signal that is similar to the amplitude of ladder curve will be coding directly. Otherwise, any analog signal not intercepted with this

curve will be mapped to the nearest ladder digit.

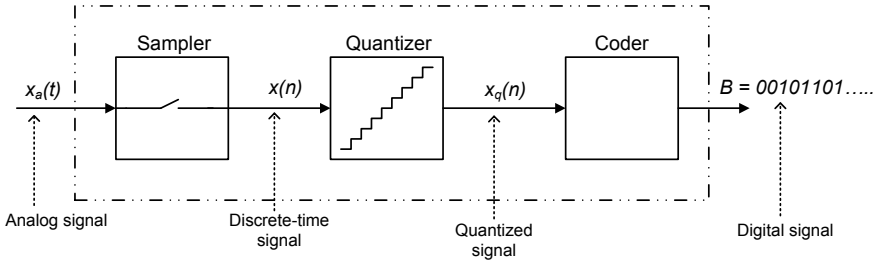


Figure 2.10: General ADC operation

An ADC is a kind of a 'passive' device that represents time-sampling analog signals as equivalent to the digital wordlength N. In this case, the ADC does not alter any information carried by the signal in the analog domain, but instead transforms the information into the digital wordlength. An N_{min} value determines the minimum detectable signal in the presence of noise. The smallest resolution value in terms of the voltage amplitude, V_{LSB} is given by

$$V_{LSB} = \frac{V_{FS}}{2^N} \tag{2.1}$$

where V_{FS} is the full-scale voltage range at the ADC input. The digital output of an ADC is determined by the sampled analog signal at the corresponding conversion. An *N-bit* ADC will have 2^N output levels. Each sample is compared with the nearest quantisation level. In this case, each level is assigned a specific code of *N-bit* length, where all zeros $(000\ldots00)$ is the lowest level and all ones $(111\ldots11)$ is the highest level. The range between the lowest and the highest levels is known as full-scale (FS) range.

2.5.1 Applications and Architectural Choices

Since digital processing offers many advantages and performs better than analog processing, the demand for ADCs has grown rapidly in the electronics industry. However, the requirements are not the same and differ from one application to

another. Therefore, ADCs are selected according to specifications and application.

Generally, an ADC is categorised based on design and architecture. Two main important criteria of selection are: i) the speed of conversion determined by f_s, and ii) the accuracy of the digital bit string represented by N. The performance of ADC is a compromise of these two parameters. Both N and f_s also determine the ADC topology, how they are used, and the complexity of the architecture.

To date, researchers have proposed a number of ADC architectures. In certain applications, response time and hence, the speed are important. Meanwhile, for other uses, resolution and accuracy are the main interests. High-resolution ADCs occupy the circuit footprint of a larger area than the low-resolution ADCs. Similarly, increasing f_s will result in higher power consumption and circuit complexity. In precision measurement and sensor conditioning, high-resolution ADCs are required [38]. Pressure and temperature sensors are examples of which signal remains unchanged for prolonged periods, with a brief burst of significant activity, and hence required low f_s. Meanwhile, high-speed and medium-resolution ADCs are used widely in wireless applications and consumer electronics [39].

Available ADC architectures are categorised in two major groups: *Nyquist* or *Oversampling* ADCs [4]. Here, the architectures are differentiated in terms of precision (N) and speed (f_s). The variation of the precision-speed curve, as presented in [40] and highlighted in Fig. 2.11, demonstrates the selected ADC architectures.

Nyquist type converters operate at specific N and f_s. An integrating converter for instance, is precise but has to be sacrificed for speed. Meanwhile, flash and pipeline converters are mainly used for very high-speed systems but provide lower resolution. The successive approximation (SAR) ADC provides a balance between N and f_s and are categorised as medium converters. This resolution-speed tradeoff is considered because of its limited power consumption, size and precision, especially for portable devices. In theory, high-speed ADC can still be developed with high resolution. However, the architecture size will be very large

and thus not suitable for mobile devices.

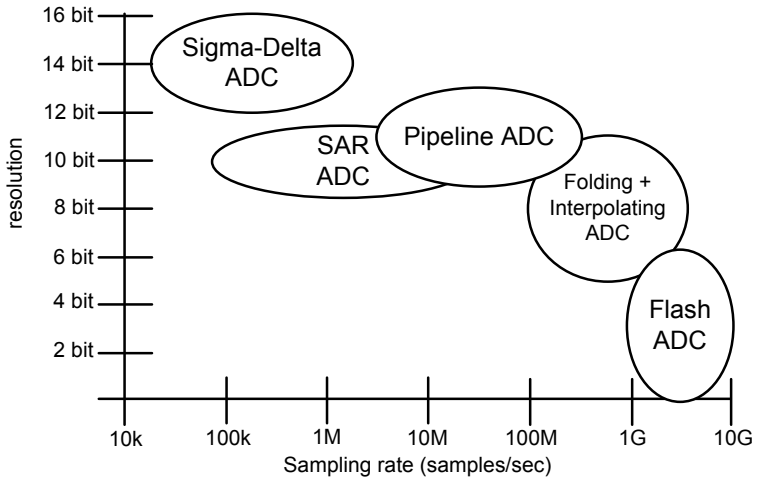


Figure 2.11: ADC types and choices: resolution vs. sampling-rate

In contrast, an *oversampling* or Sigma-Delta ($\Sigma\Delta$) type converter requires much faster and very high clock speeds to convert the input signal. In this case, it is typically 20 to 512 times faster [4]. The main advantage of such a converter is that it relaxes the demand of a high-resolution digital-to-analog converter (DAC), which converts the digital signal into analog form. In an ADC, the DAC is used in a feedback path to stabilise the converted signal, as well as to convert the analog signal accurately.

2.5.2 General Subcircuits in ADC Architecture

In system level design, the whole architecture may be divided into several functional subcircuits. The most common blocks for the ADC are the current source, sample and hold (SH) circuit, operational amplifier, comparator, and DAC. Circuit design techniques influence the performance of CMOS ADCs and this is discussed in detail in [41].

2.5.2.1 Current Source, Current Sink and Current Mirror

Current source and current sink are used widely in integrated circuit (IC) design. Both possess an identical operation except that a current source is used to draw current from a p-channel MOS (PMOS) transistor, whilst a current sink is used to supply current to an n-channel MOS (NMOS) transistor [37]. The method of using current sources and current sinks is the art of biasing in an IC design and pushes the transistor into operation.

The other most useful basic building block for the IC design is the current mirror. It uses the original idea of current source or current sink; however, it is applied with a transistor to form a self-biased circuit and a tunable current-driven circuit. The principle of operation is simple: if the gate-source potential of two identical CMOS transistors is equal, then the channel currents should be equal.

2.5.2.2 Sample and Hold Circuit

In the previous description, the sample-and-hold (SH) circuit is identified as the first stage of digitisation. It is used to sample an incoming analog signal and holds it until the conversion process is completed. To increase the dynamic range, it is designed in the differential pair input. SH is built from several sub-blocks. Fully differential amplifiers are used at the input stage. Meanwhile, a complementary switch samples the input data for every edge-triggered clock frequency. Capacitors are used to store the sampled signal for the duration of quantisation and coding. For the output stage, a high-impedance unity-gain buffer is used. The buffer is built from a common-source amplifier. This configuration is not only suitable for low speed application but is also capable of handling high-speed while keeping its supply voltage low [42].

2.5.2.3 Operational Amplifier

The operational amplifier, or opamp, is another important building block that can be found in most electronic circuits. The details of a CMOS opamp design is dis-

cussed in [43]. Basically, it is developed from a selection of single stage amplifier configurations: common-gate, common-source, or common collector amplifier circuits. In CMOS, several performance objectives for opamps have been outlined. Gain is an important measure for the opamp: either open loop or closed loop gain. The gain can be either voltage or current. DC offset, DC biasing, and the power supply rejection ratio (PSRR) are design considerations for good opamp performance. Careful design will minimise noise from internal circuitry. Frequency response, slew rate, and power dissipation are other important measures for opamps. Meanwhile, circuit compensation is used to correct any performance deviations.

The architecture of an opamp has a wide variety of uses, including topology variations. The variation in circuit configuration reflects its performance. In general, the demand for specific requirements stimulates the circuit innovations.

2.5.2.4 Comparator

A comparator performs digitisation of analog signal. The digital signal of an ADC is given by the comparator output. This block compares two signals, incoming analog input and a reference input, and produces a digital output of 1s and 0s based on the comparison. If the signal input is larger than the reference signal, the comparator gives a HIGH output (1). On the contrary, when the signal is smaller than the reference signal, a LOW output (0) is generated [37].

Apparently, the comparator is based upon an extended opamp circuit, in which the obvious topology has a two-stage differential input or a cascade of AC coupled self-biased inverters. However, both of these topologies have problems of charge-pumping and poor PSRR. To overcome this problem, the authors in [44] proposed an architecture of fully differential variable supply comparator.

Often, the offset of comparators limits the resolution of ADCs [45]. A topology of a chosen comparator is based upon design constraints and selected architecture with offset compensation and optimisation. For the output stage, a source fol-

lower is used together with latches for temporary storage. The authors in [46] demonstrated a CMOS comparator for medium resolution ADC implementation using an offset-cancelled amplifier to reduce input offset.

2.5.2.5 Digital-to-Analog Converter (DAC)

ADC uses DAC in its feedback path. In many ADCs, accuracy and precision of the resolution rely strongly on the performance of the DAC; sometimes referred to as a subDAC, since it is the integral part of the ADC. A basic diagram of the DAC is shown in Fig. 2.12, in which a string of digital inputs is converted to a continuous analog output signal.

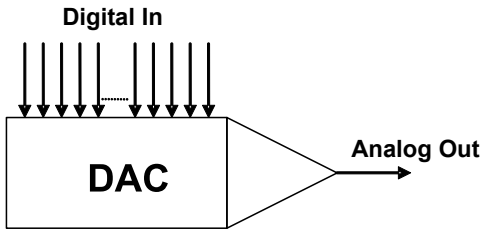


Figure 2.12: DAC block diagram

There are many DAC types available, ranging from decoder-based converter, binary-scaled converter, thermometer-code converter to a hybrid converter [4]. Decoder-based architecture, such as a resistor string converter, requires large die area and is not suitable for integrated circuit applications. Similarly, a hybrid converter also integrates some resistor elements in its architecture. Thermometer-code converter, on the other hand, requires many cycles $(2N - 1)$ to convert the *N-bit* string. Therefore, in moderate and fast integrated circuit applications, a binary-scaled converter is more feasible since an array of CMOS current sources can be used to implement high-speed current-steering DAC, as demonstrated in [47].

2.5.2.6 The Digital Circuit

A small part of an ADC consists of the digital circuit, for coding, error-correction, calibration, output register, modulator, or decimation processes. A digital output buffer is included to hold the data during transmission. The most common ADC encoding schemes are straight-binary, offset-binary, and two's complement [48]. The decoder is one factor that dominates the performance of a flash ADC [49]. Meanwhile, digital calibration and correction are used widely in a pipeline ADC to improve its precision due to component linearity and mismatch [50,51]. Similarly, a SAR ADC requires a digital register as a part of its operation with the addition of digital calibration [52, 53]. Likewise, $\Delta\Sigma$ ADCs consist of digital modules of modulator and decimation filter to operate [54, 55].

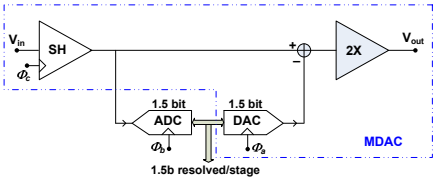
2.5.3 Pipeline ADC

A pipeline ADC is used in a wide range of applications [40]. It is also a popular choice for wireless applications [56] due to its advantage of high-speed operation and its provides a sufficient range of resolution for such applications.

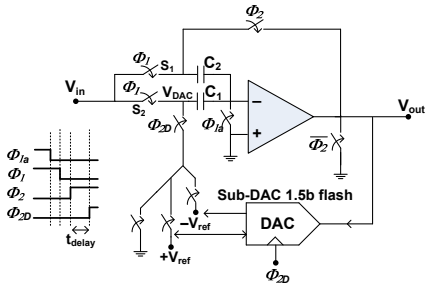
A single pipeline stage consists of sub-ADC and multiplying-DAC (MDAC), as illustrated in Fig. 2.13(a). The MDAC, as shown in Fig. 2.13(b), comprises of a SH, an amplifier, and switched-capacitor circuits [57]. In this case, a single amplifier is used for sampling, holding, and multiplying stages, and different switching configurations are applied for each stage. Usually, a 1.5-*bit* sub-ADC is used, which uses two comparators to generate the 2-*bit* output, b_0 and b_1, for each pipeline stage, as shown in Fig. 2.13(c). Meanwhile, a binary-scaled DAC is used to convert the sub-ADC output into analog for residue calculation. Figure 2.13(d) presents this 1.5-*bit* sub-DAC, consisting of an amplifier and binary-weighted switched-capacitor network.

The number of stages determines the N value of pipeline ADC, where each stage effectively is equivalent to 1 *bit*. Each stage is cascaded together, as the first stage is equivalent to the most significant bit (MSB) and the last stage is the least

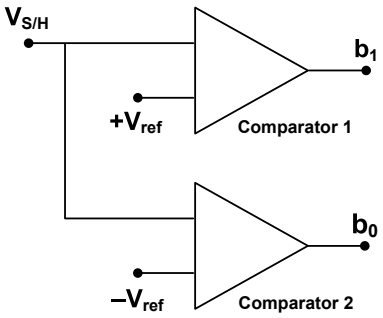
significant bit (LSB).



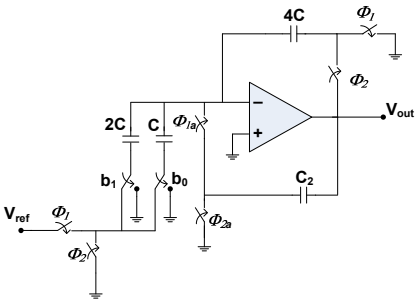
(a) Single-stage pipeline



(b) MDAC



(c) 1.5-*bit* ADC



(d) 1.5-*bit* DAC

Figure 2.13: Circuitry of pipeline stage

2.6 Low-Power Design

2.6.1 CMOS Process Technology

Digital circuit design is implemented in CMOS process technology because the downscaling process ensures small footprint architecture. Previously, analog designs widely used the bipolar process because of its better current gain. However, due to high demand for low-power products and serious attention towards a single chip solution, CMOS processes mostly dominate the current research activities of analog circuit design.

The rapid development in CMOS downscaling is a healthy scenario for low-power design. Deep submicron (DSM) technology reduces the magnitude of supply

voltage and hence, is suitable for low-power devices. Due to its superior shrinking size, system integration will ensure the dimension and product size is small and lightweight, which is suitable for portable devices. As a precaution, the integration has to take into account issues of performance, cost of design and manufacturing, power consumption, and reliability. However, this downsizing process will come to the end when it can no longer be reduced. The ultimate limit for downsizing is equivalent to atom distances in the crystal lattice and is approximated to 0.3 nm [58].

Another related issue in CMOS is its technology characterisation for analog circuit design. There are six fundamental characterisations to be considered: DC behaviour, AC behaviour, linearity, matching, temperature dependence and noise issue [59]. Each parameter contributes to circuit performance. Furthermore, there is a trade-off between the parameters together with speed, gain, impedances, power dissipation, and voltage issues. Thus, in realising a CMOS analog circuit design, one must be aware of the existence of performance-dependent parameters to design the module according to specifications accurately.

2.6.2 Circuit Design Approach

Low-power circuit designs are vital for portable devices and battery-operated applications. Apart from using a CMOS process that operates with a low voltage, a circuit design approach also plays an important role. An obvious method is to switch functional circuits ON and OFF according to the operation modes. In this case, unused blocks are kept in idle mode when not in use.

Two other techniques can be applied in circuit design to keep the power supply low: adaptive biasing and manipulation of the subthreshold region of CMOS transistors.

In adaptive biasing, an input dependent bias current is used to reduce the power consumption by having an extra precaution on mismatch effect and stability. The amplifier increases the bias current with the presence of input signal.

Otherwise, the operation remains at a very low current [60]. Eventually, the use of this dynamic amplifier approach will increase the slew-rate of the amplifier. Likewise, an ultralow-power design can also be achieved with the self-biased current source (SBCS) that easily be reused [61].

The operation of the transistor in the sub-threshold region introduces another choice for low-power design. This eliminates the requirement of a threshold voltage V_{TH} necessary for the transistor to operate in the saturation region as in the normal CMOS design approach. The authors in [62] and [63] demonstrated the use of sub-threshold region in designing a low-power amplifier. These included manipulation of transistor operation in weak and strong inversion.

The current source is the most important element in integrated circuit design and is applied widely to bias the circuit. Therefore, it is equally important to apply a low-voltage CMOS current source in ADC design to minimise consumption, as implemented in [64].

2.7 Power Evaluation for ADC

2.7.1 Overview

Power consumption has become a critical issue in a battery-operated wireless device. Since batteries must be small and light to reduce the total weight of devices, current consumption must also be limited. Designing an electronic system for wireless is a tedious task. It involves multiple disciplines, expertise and knowledge, such as analog, digital, radio-frequency, communication, and so on, as well as different design layers, such as physical, datalink, and networks. There are also various wireless applications, such as communication, sensors, multimedia, and medical. Since the scope is very wide, it is critical to estimate the power consumption of a device at a system level to optimise power usage and assists the design process simultaneously.

ADC consists of several operational blocks, such as comparator, amplifier,

DAC, SH, and CMOS switches. The arrangement of these blocks is according to specific topology. In Nyquist ADC selection, the commonly used architectures are flash, pipeline and SAR, algorithmic, and cyclic ADCs. In each topology, a digital unit is also required for calibration, coding, or as an output buffer.

2.7.2 Power Consumption of ADC

Power consumption of a fixed or conventional ADC can be estimated both at the system and circuit levels. At the system level, only the ADC information of N and f_s are necessary. In this case, the power consumption is evaluated based upon N and/or f_s of an ADC. With more detailed information of circuit design parameters such as supply voltage V_{dd}, amplifier gain A_v, CMOS minimum channel length L_{min} and bias current I_{bias}, a better estimation may be obtained at the circuit level.

2.7.2.1 System Level Estimation

Power consumption of an ADC is limited by technology scaling and its internal analog sub-circuits, including circuit optimisation [5, 65]. In general, the power consumption of an ADC is estimated by (2.2) and (2.3); the graph of the power consumption model at the system level as described in [66] is shown in Fig. 2.14.

$$P_{ADC} = \frac{kT_K}{t_s} 10^{\frac{6N+1.76}{10}} \quad W \tag{2.2}$$

$$P_{ADC} = k.T_K.f_s.10^{(6N+1.76)/10} \quad W \tag{2.3}$$

where $f_s = \frac{1}{t_s}$, where k is the Boltzmann's constant, T_K is absolute temperature in Kelvin, N is the ADC resolution, and t_s is the sampling interval.

According to (2.2) and (2.3), low f_s and a minimum value of N reduce the power consumption of an ADC (P_{ADC}). In this case, P_{ADC} is directly proportional

to f_s and varies logarithmically with N. Applications that use ADC at low N might show small power changes. However, P_{ADC} is significant when high N is required.

As estimated by the authors in [67, 68], the total power consumption of an ADC for CMOS Nyquist-rate high-speed ADCs yields

$$P_{estimate} = \frac{V_{dd}^2.L_{min}.(f_s + f_{sig})}{10^{-0.1525.ENOB+4.838}} \quad W \tag{2.4}$$

where f_{sig} is the signal frequency, and $ENOB$ is the effective number of bits.

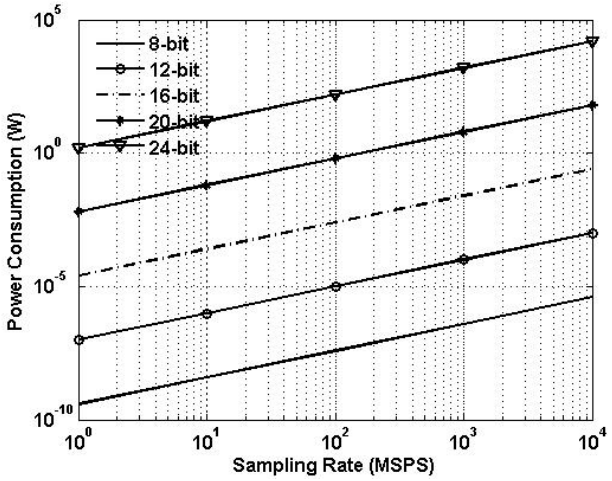


Figure 2.14: ADC power consumption for different f_s and variable N values [66].

2.7.2.2 Circuit Level Estimation

Without detailed knowledge of the circuits and by considering analog design parameters, the ADC power estimation is given by [69]:

$$P_{ADC} \propto \sum \frac{1}{2}.\frac{A_v}{k_\alpha}.V_{eff}.V_{dd}.L_{min}.f_s \quad W \tag{2.5}$$

with $V_{eff} = V_{GS} - V_{TH}$, where V_{GS} is the voltage between the gate and source

of a CMOS transistor and V_{TH} is the threshold voltage. Here, k_α is a coefficient according to specific analog circuits, such as SH, comparator, gain amplifier, etc. In both equations (2.3) and (2.5), the power consumption is proportional to f_s. In (2.3), k and T_K are constants whereas in (2.5), V_{eff}, V_{dd}, L_{min}, A_v, and k_α are optimised at the circuit level. Thus, by varying f_s at the system level, the power consumption will also change.

In summary, according to [69], the power consumption of an ADC at the circuit level can be estimated as:

$$P_{estimate} = \frac{\rho_i.V_{dd} - \sigma_i.V_{swing}}{\eta_i}.V_{dd}.L_{min}.f_s \quad W \tag{2.6}$$

where ρ, σ, and η vary for different building blocks.

By considering circuit elements, Table 2.2 summarises the power approximation for the most used ADC types, such as flash, pipeline, SAR, and $\Sigma\Delta$ according to [69].

2.8 Design Flow

The design flow, as depicted in Fig. 2.15, describes the design process adopted in this thesis. In this work, the design starts with studying the standard document specification and continues up to the place and route (PNR) stage. Firstly, the specific requirement for the ADC parameters is obtained from standard UMTS/FDD documentation. Subsequently, a specific ADC topology is selected. The behaviour of the ADC is observed to introduce a suitable reconfigurable mechanism.

MATLAB simulation is used to validate the proof of concept of the proposed adaptive algorithm. At this stage, various conditions and solutions can be tested extensively since the algorithm can be modified and verified quickly. Once a specific solution is achieved, test-benches are created to assist hardware verification

Table 2.2: Power estimation for different ADC topologies

ADC architecture	Power Estimation
Flash	$P_{flash} = (2^N - 1).P_{comparator} + P_{encoder}$
Pipeline	$P_{pipeline} = N_{stage}.(P_{SH} + \frac{(P_{ADC} - P_{gain}).N}{N_{stage}} + (2^{\frac{M}{N_{stage}}} - 1).P_{comparator}) + P_{DEC}$ $P_{pipeline} = N_{stage}.(P_{SH} + \frac{(P_{ADC} - P_{gain}).N}{N_{stage}} + (2^{M/N_{stage}} - 1).P_{comparator}) + P_{DEC}$
SAR	$P_{SAR} = \frac{N}{m}.(P_{SH} + m.P_{DAC} + (2^m - 1).P_{comparator}) + P_{register}$
$\Delta\Sigma$	$P_{\Delta\Sigma} = R_{oversample}.(M_{order}.P_{integrator} + P_{comparator} + P_{DAC}) + P_{decimation}$

throughout the design flow. It is important to use these 'standard' test-benches to ensure correct performance at each design stage.

The successfully verified algorithm is then coded by Verilog hardware language and simulated using Verilog XL form Cadence Design System.

Hardware coding is mapped into a technology dependent library using the Synopsys Design Compiler once the coding is verified. In this work, a technology library of $0.18\mu m$ from UMC is used. The gate level synthesis file is then verified using the standard test-benches generated from MATLAB. At this stage, the switching activity interchange format (SAIF) file is extracted. This file is used by the Power Compiler to calculate the power consumption after post-synthesis simulation.

The post-synthesis gate level evaluation gives good power estimation. However, this estimation is based on wire load models provided by the technology library. Post-layout power evaluation is necessary for a more accurate estimation. In this case, more accurate interconnect capacitance is extracted from the layout. An automatic PNR tool, Silicon Ensemble, is used to obtain the layout file.

Power consumption can be monitored at different design stages, either after synthesis or after PNR. Throughout the design process, different architecture and circuit configurations are tested until the desired performance is reached. For fast comparison between solutions, post-synthesis power estimation is sufficient. At the final stage of the design, when more accurate results are required, the post-layout power estimation is used.

2.9 Summary

In summary, ReHW is important for electronic system designs due to its many advantages. With portable devices dominating the market, the requirement to stretch trade-offs between speed, power, performance, reusability, flexibility, granularity and computation efficiency becomes crucial. The reconfigurable approach is not only helpful during the design stage but could overcome many post-production

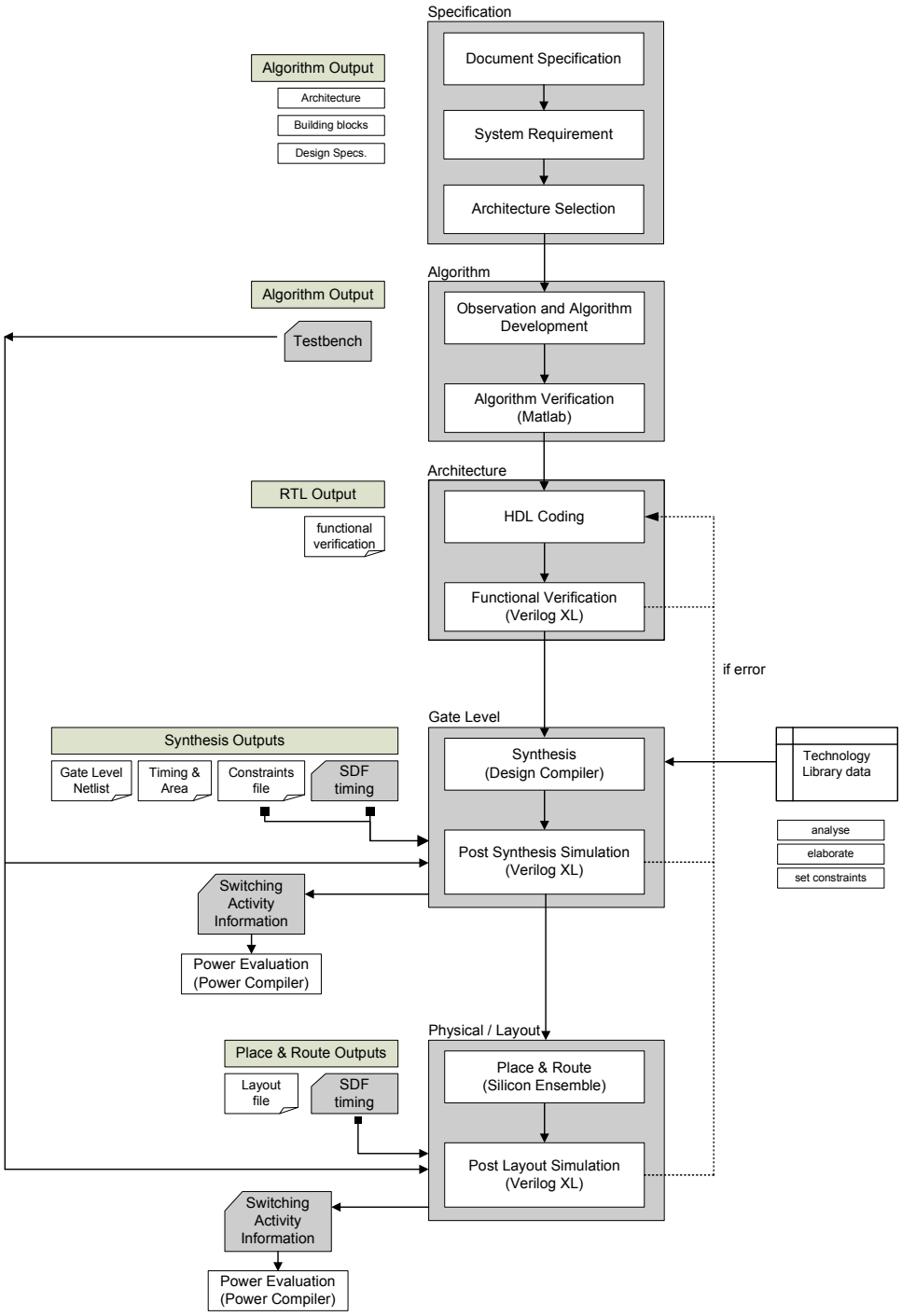


Figure 2.15: Design flow for hardware implementation, verification and power evaluation

problems and therefore enhance the dynamic behaviour of a system. Undoubtedly, ReHW is a strong contender to solve multiple applications and multi-mode operation problems for portable devices. However, designing a reconfigurable system introduces additional design issues that should be examined carefully and rigorously before such a system can be implemented successfully.

Recent advancement in wireless technology has enabled high data-rate transmission and improved QoS. Meanwhile, the selection of transceiver topology is highly influential to the performance of wireless devices. Current trends in electronics are targeting compact and low-power implementations. Hence, apart from using improved circuit techniques, the choice of transceiver topology for wireless devices is equally important.

Data conversion is the crucial interface between the real and digital world processing system. ADC is a key conversion device used at the interface between analog and digital domains. In modern digital systems, ADC is essential. For wireless communication, designers of ADCs, especially for the Nyquist-rate converter, are facing critical challenges in N and f_s. As the frequency increases, dynamic range requirements also increase.

It is important to study the building blocks in ADC architecture, to understand their operations, and to classify the factors that affect the ADC performances. Once the contributing factors have been identified, improved circuit techniques can be implemented.

Power consumption issue of an ADC is very significant for battery-operated applications and portable wireless devices. It is important to draw only small amount of power from the supply to prolong battery lifespan. The device lifetime could also be extended by keeping the heat dissipated by the module as low as possible since highly dissipated power can overheat the device. To move towards a green environment, it is crucial to reduce the power consumption of electronic devices. It is also equally crucial to continuously improve the performance of the devices and simultaneously keeping the power consumption at a minimum.

The next chapter will discuss the existing research works on ReADC as a new technique to increase adaptability and the performance of the converters for future wireless applications.

Chapter 3

Review on Reconfigurable ADCs

Contents

3.1 Introduction

With an increasing need for low-power schemes in green technologies, engineers have turned to reconfigurable design methodologies. An ADC, the most common unit in modern digital devices, has become a target module for adaptively varying its N and f_s. This has led to the scaling of its energy consumption to reduce power intake.

Extensive research work on algorithm, circuit design, analysis, and prototyping on ReADCs has been carried out in the last decade. This chapter will review the development of ReADC research, especially regarding the aspect of applications and architectural topologies. Configuration methods are also discussed. Specific

attention to the development and the state-of-the-art of ReADCs for wireless communication and mobile devices is also included. Particular emphasis on the adaptive controllers for the ReADCs will be given.

3.2 Review of ReADC Architectures and Applications

In this section, a review on the previous research studies regarding architectural ReADCs and their applications is provided. Recently, there have been a few research studies on ReADCs. These works include design, development, and prototyping of ReADCs in various applications, architectural topologies, and process technologies.

Each of these processes requires different approaches and architectural configurations. Nevertheless, research in this field is quite new and hence published research papers are still limited. Therefore, there is room for the exploration of different architectures, circuit configurations, reconfigurable approaches, design methodologies, and strategies. In addition, applications, design environments, and implementations of ReADCs offer many design improvements.

In 1996, the development of ReADC was implemented using the FPAA module [70]. In this study, different circuit configurations were achieved between ADC, DAC, and SH circuits. A complete work comprising FPAA and digital controller using FPGA was subsequently discussed in [71]. This work was the first attempt on ReADC development. The authors emphasised the concept of reconfigurable methodology and the developed module was not specified to any application. The latest development of ReADC implemented using FPAA is reported in [72]. This architecture is based on continuous-time $\Sigma\Delta$ architecture developed using $90nm$ technology. The architecture consists of a hexagonal structure FPAA to configure the filters.

ReADCs based on $\Sigma\Delta$ [73], pipeline [74], flash [75], hybrid pipeline-$\Sigma\Delta$ [76],

and hybrid flash-SAR [77] for general application are also reported.

An interesting study on ReADC architecture expressed three configuration methods: architectures, parameters, and bandwidth reconfigurations [78]. In architecture configuration, the proposed method was able to switch between two different topologies: pipeline and $\Sigma\Delta$. For fast applications, the pipeline converter was used, while $\Sigma\Delta$ was selected for high resolution. These architectures were not completely different in terms of their functional sub-blocks. Both topologies shared common circuits, such as comparator and DAC. The difference was only a few subcircuits, such as the integrator, that was used in $\Sigma\Delta$ configuration but not in the pipeline mode. Each architecture could also change its behaviour. For example, the pipeline converter could change its length of the pipeline arrays. In $\Sigma\Delta$ mode, the oversampling ratio (OSR) could be varied for different resolution levels. For both architectures, the size of capacitor could be changed for fine-tuning of N. Thirdly, PLL was used to measure the input bandwidth and changing the biased current to optimise power usage according to the different operation modes. When a higher resolution mode was needed, more bias current would be drawn into the circuit.

For mixed-signal applications, a ReADC of hybrid pipeline, cyclic, and time-interleaved was proposed in [79]. With a similar concept to [70], granular FPAA was used as the architecture base to achieve reconfigurable resolution (reN). Meanwhile, a pipeline ReADC with reconfigurable sampling-rate (reFs) for same application was discussed in [80].

A flash ReADC was presented in [81] for the digital servo application. The proposed converter was targeted for digital read channel applications, and N was changed between 6 *bits* and 7 *bits* to achieve fast and slow operational modes.

Another proposed architecture was the two-mode ADC reconfiguration design [81]. The converter could operate in fast, low-N mode flash-like ADCs and in slow and high N using interpolating latches. Meanwhile, due to their high precision requirement, SAR and algorithmic ADCs are still used commonly in

sensor applications. Reconfigurable SAR ADCs for sensor networks with different N [82] and f_s values [83, 84] were reported. For miniaturised and autonomous sensor applications, a hybrid algorithmic-$\Sigma\Delta$ sensor was presented in [85]. Quite recently, a reconfigurable flash was developed for image-sensor applications targeting low-power electronics [86].

Meanwhile, for smart sensor and data acquisition implementations, ReADCs based on pulse width modulation (PWM) were developed [87]. The modules used a micro-controller/PC to program the ADC digitally to encounter various transducer nonlinearities.

For mechatronic applications, the authors in [88] proposed a $\Sigma\Delta$ ReADC. The converter was implemented in Xilinx Virtex4 FPGA, and its filter configuration can be altered. Real-time application, however, was limited due to its complex specification.

Biomedical applications, including bio-acquisition systems and electroneurography have also become interesting areas of research for ReADCs. Converters based on $\Sigma\Delta$ [89], pipeline [90], and SAR [91] have all been developed. ReADCs can also be used in fault-tolerant and testing applications [92]. Meanwhile, to reduce power consumption in multimedia applications, such as high definition television (HDTV) and standard definition television (SDTV), a pipeline ReADC with variable f_s was presented in [93].

In wireline applications, which require fast data rates up to the gigahertz (GHz) range, a time-interleaved ReADC has recently been developed; it uses an externally-driven periodic signal to configure the ADC's counter [94].

A model of voice-codec mixed signal front-end transmission was described using a microphone, ADC, digital signal processor (DSP), DAC and speaker in [95]. A $\Sigma\Delta$ modulator was used with an oversampling ratio of 512. The circuit was designed using a 0.18 μm digital CMOS process supplied at $1.8\mu m/3.3V$. A ReADC stage was implemented both in analog and digital blocks, taking advantage of the half-duplex operation and reducing the total area and power consumption.

Circuit switching is a widely used method for ReADC in the mentioned literature. A switched-capacitor (SC) circuit is implemented to change: (1) architecture, (2) routing, (3) circuit selection and behaviour, and (4) parameters values that use reconfiguration data.

3.3 Review of ReADCs for Wireless Communication

Wireless technology is computationally intensive and requires flexibility [96]. For portable mobile devices, a flexible design scheme helps to speed up the time-to-market during the design phase and manage hardware resources and energy utilisation during operation. However, reconfigurability issues are one of most important challenges in wireless systems, especially for the design of RF and baseband [27].

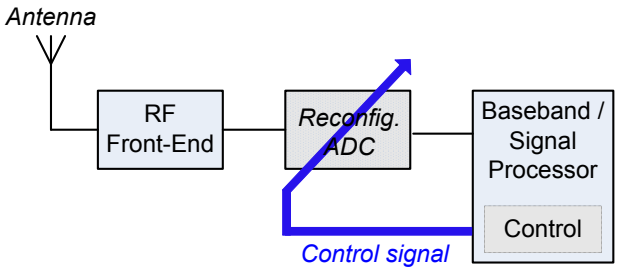


Figure 3.1: Wireless receiver front-end with ReADC

Figure 3.1 illustrates the possible use of a ReADC in the wireless receiver front-end. The adaptive system, which is a closed-loop configuration, consists of a configurable ADC unit, a digital processing unit, and a control unit. The output of the ADC is being processed and measured continuously. The output of the measurement system is fed through a controller unit to decide on the changes. In most cases, the controller unit consists of memory cells that compare the output of measurement unit and the desired values. Later, the control unit re-routes the switches inside the ReADC according to the desired conditions. In general, the

configuration of the ReADC involves two main parameters, N and f_s, and both parameters can either be increased or decreased.

To date, several researchers have worked on ReADCs design at the system level [97], architectural level [98], or circuit level [99]. For wireless communications, such as UMTS, Bluetooth, WLAN, WiMAX, WiFi, UWB, RFID, GPS, and 4G, either Nyquist-rate fast converters or oversampling $\Sigma\Delta$ are normally selected [100–108]. It has been demonstrated that pipeline converters [109, 110], flash [111], $\Sigma\Delta$ [112], time-interleaved (TI) [113], continuous-time (CT) $\Sigma\Delta$ [99], and discrete-time $\Sigma\Delta$ [114] can be used in wireless applications. Meanwhile, the hybrid ADC architecture strategy of pipeline-cyclic [115], pipeline-$\Sigma\Delta$ [108], flash-$\Sigma\Delta$ [116], and TI-SAR [117, 118] are used for the multi-mode operation.

The authors in [119] proposed a pipeline ReADC. It specifically targeted the UMTS receiver. The main advantage of this architecture is its precise control of N, which may be varied from 4 *bits* to 16 *bits*. This architecture was based on a previously developed scalable topology by the same authors [120]. The authors then extended the implementation of the architecture for a UMTS time division duplex (UMTS/TDD) mode for the cellular communication [121]. Meanwhile, the authors in [122] introduced a wireless architecture that supports both GSM and UMTS applications using a reconfigurable $\Sigma\Delta$ modulator.

Authors in [123,124] have published specific implementations of a ReADC for UMTS applications, including adaptive strategy . Pipeline ReADC was implemented in MS, which changes N due to its adjacent channel interference.

In summary, all these works focused entirely on the architecture and the circuit techniques for ReADCs, assumed to be triggered by a control signal from the baseband processing unit. Detailed discussion of the adaptive feedback controller mechanism, such as measurement unit and reconfigurable control units, is very limited.

3.4 Review of Control Schemes for ReADCs

Whilst an extensive discussion of programmable ADC architecture for mobile communication can be found, the configuration mechanism described by authors in [78,97,108,124,125] is triggered by external control signals generated either by using DSPs or PC.

Anderson et al. [22] mentioned the control circuit, however, no implementation was found. Meanwhile, the adaptive line enhancer (ALE) for noise cancellation was implemented in [103] to boost the SNR, but such implementation can only be restricted to $\Sigma\Delta$ ADC.

Stojcevski et al. reported on pipeline ReADC for UMTS applications in various publications [121,123,124,126,127] with the main topic focusing on reN, which is used to suppress the interference signal by measuring both signal and interference power levels. Both channel and pulse shaping filters were implemented in FPGA [128]. The filters were required as a part of an adaptive algorithm implementation that used DSP [129] for controlling the ReADC configuration. The controller monitored the adjacent channel interference and adjusted the converter wordlength automatically to reduce the power dissipation. However, the feedback control mechanism involved several other blocks, such as a pulse-shaping RRC filter, a downsampler, a de-scrambler, and a de-spreader. In this implementation, the decision process required much longer time, since a filter and a DSP unit had processed the digital output from the ReADC, before the controller could adjust the configuration switches. This reconfiguration method had also taken a longer data-path, as well as occupying a larger area than using a conventional filter to perform such a function. Furthermore, the implementation required both a reconfigurable digital filter using FPGA and a controller unit using a DSP. Implementation of the adaptive algorithm required an output signal from the filter. The DSP was included to perform configuration processing that sent the signal to the pipeline ReADC unit. The filter occupied an area complexity of 15000 NAND equivalent gates/mm^2 and consumed as much as $52mW$ in $0.35\mu m/2.7V$ CMOS

process. Meanwhile, the DSP for adaptive algorithm implementation consumed a fixed power of $1.7W$ [130]. Hence, implementation of controllers using DSP certainly consumes more power than ASIC.

3.5 Discussion

The latest developments in ReADCs have focused on fast and low-power converters. To achieve low-power performance, the circuits are designed carefully with various circuit design approaches. Similarly, nanometer scale CMOS technology is used for prototyping. This has resulted in low operational voltage and hence, low-power consumption. Meanwhile, the current design trend to achieve fast operation relies on the hybrid implementation of TI architecture with other various ADC types, such as pipeline, SAR, flash, and $\Sigma\Delta$.

Until now, the existing works and state-of-the-art design of ReADCs have concentrated on architecture types and applications. Most of the developed ReADCs have focused entirely on analog circuit techniques. While the details of the circuit configurations are elucidated extensively, the discussion on configuration scheme is very limited. External control signals from the baseband are assumed to trigger the reconfigurable hardware. The control signals are generated after intensive digital operation by the DSP.

The discussion on adaptive mechanism for ReADCs is also limited. Although, when an adaptive algorithm is presented, its implementation is restricted to the DSP core due to algorithm complexity. For reN ADCs, no discussions on the effect of changing ADC wordlength to the baseband were found. Thus far, signal-dependent detection is restricted only to asynchronous ADC for low frequency applications [131]. In this case, however, the converter is non-adaptive. Despite numerous journal and conference papers published until the present, an adaptive strategy for ReADC using information from analog waveform variation has yet to be found.

Therefore, the review of the works has given an opportunity to explore a potential adaptive control scheme that enables the ReADC to be configured according to the variation of the analog signal without the need of the DSP or baseband processing. In this case, the controller monitors the internal behaviour of the ReADC and adjusts its parameters (N and f_s) adaptively to suit the current signal behaviour. Consequently, this will have a significant effect on power consumption. To achieve a single chip solution, the adaptive mechanism is required to be as simple as possible so that the implementation in ASIC is viable. The implementation of the digital controller in ASIC will reduce the overall power consumption as compared with the existing digital controllers that have been realised using DSPs and consequently, will minimise the reconfiguration overheads utilised by the controller. The feedback path is restricted to be as short as possible where the controller should be close to the main ReADC units to ensure fast response. Furthermore, power consumption and area complexity of the adaptive controller unit should be minimised as compared with the core ReADC units.

In wireless applications especially for mobile phones, a pipeline ADC is selected due to medium to fast conversion. Moreover, pipeline architecture is regarded as a suitable candidate for ReADC in battery-operated devices due to its granularity nature in hardware architecture. This will enable the power consumption to be scaled linearly depending on its internal circuit operation, ensuring efficient power management.

3.6 Summary

ReADCs will be important for future wireless applications. With the advantages of performance improvement, scalable power utilisation, efficient computation, and dynamic hardware re-use, they will benefit the battery-operated wireless devices in modern digital electronics significantly. The main reconfigurability issue in the adaptive method is the ‘detect-and-change’ mechanism. In real-time ReADC

(adaptive ADC), discussion on this issue is very limited.

This chapter presents existing adaptive techniques for ReADCs for mobile applications. The chapter begins with a review of relevant ReADC studies, with different applications and circuit topologies. Furthermore, specific attention is given to the research papers on ReADCs for wireless communication. In addition, existing adaptive schemes are also described.

There is an opportunity to control the ReADC adaptively using information from its internal behaviour according to different amplitude variations, which is implemented directly at the analog front-end. To further investigate this potential research work, pipeline ADC is selected due to its granularity nature that enables linear power-scaling ReADC implementation. The digital controller will be implemented as close to the main ADC module as possible to achieve a single chip solution. To reduce the power consumption, the controller will be implemented in ASIC, with behaviour derived from an adaptive algorithm.

In conclusion, there is much potential research exploration in an adaptive control scheme due to limited discussion on the design, reconfiguration approach, and its implementation. At the moment, the adaptive controllers and configuration instructions are implemented in the baseband. Furthermore, the reconfiguration approach, especially the change in wordlength, is yet to be discussed. Similarly, the performance metric for ReADCs, especially regarding the reconfigurable overheads in terms of power and area for the digital controller, has not yet been found.

Chapter 4

ADC Specifications for UMTS Zero-IF Receiver Architecture

Contents

4.1 Introduction

Design trends in wireless Rx architectures for future mobile communication have pushed the digital domain closer to the antenna end. Consequently, specifications for ADCs must meet stringent requirements, according to standard wireless communication regulations. This study aims to find minimum N and f_s values for an ADC in UMTS applications. A mathematical analysis method is presented to derive the required specification for an ADC, which can be used in a Zero-IF UMTS Rx architecture targeting a MS application. UMTS standard specifications are used to derive the design plan for a Nyquist converter type.

4.2 Specifications for UMTS/FDD Mode

This section describes general UMTS specifications. Transceiver specifications described in [132] are used to explain the boundary for the Rx chain, which is the main concern for ADC designers.

4.2.1 Processing Gain and E_b/N_0 for UMTS

The UMTS/FDD standard describes a number of test scenarios for which the information bit rate R_b is fixed at 12.2 kbps and the BER must be below 0.1%. To obtain this BER, a minimum SNR is required at the Rx output. However, in the communication field, the ratio E_b/N_0 is widely used instead of SNR. The E_b/N_0 specifies the ratio of energy per information bit E_b and the noise spectral density N_0. In the case of spread spectrum technology, where R_b is protected with a chip rate R_c redundancy and coding mechanism, the relationship between E_b/N_0 and SNR is given in [32] as follows:

$$E_b/N_0 = SNR + G_p \quad dB \tag{4.1}$$

Here, G_p denotes the processing gain, which is a combination of spreading gain G_s and coding gain G_c. The spreading gain result is obtained from the spreading of a symbol rate R_s using an orthogonal pseudo-noise (PN) sequence over a wider bandwidth. This PN sequence of 3.84 MHz is known as chip rate R_c. At the Rx end, the desired signal energy is concentrated in the bandwidth that corresponds to a channel symbol rate R_s of 30 ksps during the despreading process. The symbol rate is achieved by considering a half coding rate of 60 kbps of dedicated physical data channel (DPDCH). Initially, a 12.2 kbps information bit rate is converted into a DPDCH data rate using a cyclic redundancy check (CRC), convolutional coding of 1/3 and rate matching. As a result, the SNR in the bandwidth of the despread data is improved with G_s [32], given by the following expression:

$$G_s = 10.\log_{10}(R_c/R_s) = 21 \quad dB \tag{4.2}$$

Convolutional coding adds extra bit protection to the data, and this additional protection improves data immunity. In this case, an additional 4 dB improvement is obtained from signal convolutional decoding, given by the ratio of the symbol rate R_s to the original bit rate R_b:

$$G_c = 10.\log_{10}.(R_s/R_b) = 4 \quad dB \tag{4.3}$$

Subsequently, the total improvement in SNR as a result of spreading and coding gain is

$$G_p = G_s + G_c = 25 \quad dB \tag{4.4}$$

For UMTS applications, the requirement of BER should be less than 0.1% [30]. For this amount, the E_b/N_0 is determined by simulation in [133] and should be about 5 dB. However, in reality, 1 dB to 2 dB implementation margin is necessary to ensure effectiveness, leaving the minimum requirement approximately as follows:

$$E_b/N_0\ (eff.) \approx 7 \quad dB \tag{4.5}$$

4.2.2 Crest Factor

The crest factor, CF specifies the ratio between peak amplitude of instantaneous power distribution, P_{peak} and the mean value of the input power, P_{ave}. This is given in [134] as follows:

$$CF = P_{peak} - P_{ave} \quad dB \tag{4.6}$$

The crest factor is related mainly to the Tx part and is important in designing the RF power amplifier. However, due to imperfections in transceiver chains, such as

the duplexer and filters, a portion of the transmitted signal power may appear in the Rx path. Moreover, as more signals are multiplexed in a single channel, the higher the crest factor will become. Thus, these signals may experience significant amplitude variation [135]. Depending on the number of code channels and selected codes in the UMTS/FDD mode, the CF can vary between 4.5 dB and 11 dB [134].

4.2.3 Minimum Sensitivity and Maximum Input Level

The figure of merit (FOM) for the Rx chain, as shown in Fig. 4.1 that comprises of the down-conversion process, is specified as the noise figure NF. The NF defines the ratio of SNR at the input (SNR_i) to the SNR at its output (SNR_o) of the Rx chain.

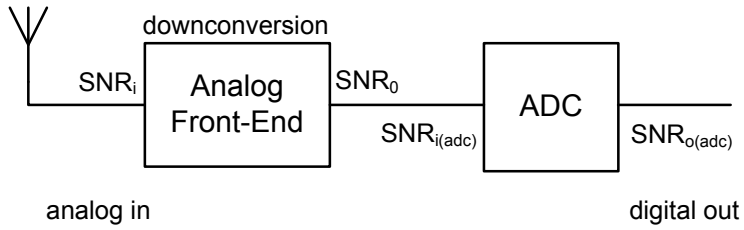


Figure 4.1: Typical Rx architecture

$$NF = SNR_i - SNR_o \quad dB \tag{4.7}$$

A typical value for the NF in a UMTS/FDD mode is 9 dB [32]. The SNR_i is specified by the difference between the reference sensitivity level, $DPCH\_E_c$ and the actual noise N_0 at bandwidth BW at an absolute temperature T_K. $DPCH\_E_c$ is the average energy per chip of a dedicated physical channel (DPCH) and is the minimum received mean power at the Rx antenna port for which the BER may not exceed 0.1%. On the other hand, the value of N_0, is given by $10.log_{10}(kT_KBW)$.

$$SNR_i = DPCH\_E_c - N_0 \quad dB \tag{4.8}$$

Meanwhile, the SNR at the ADC input is given by the ratio of $DPCH\_E_c$ and the maximum acceptable noise n_{max}.

$$SNR_o = SNR_i(ADC) = DPCH\_E_c - n_{max} \tag{4.9}$$

In [132], the minimum reference sensitivity, $DPCH\_E_c$ is specified as –117 dBm and N_0 is calculated as –108 dBm. Hence, by considering (4.1), (4.4), (4.5), and (4.9), the acceptable noise level at the ADC input, n_{max} is –99 dBm. From (4.5), the bit energy of the received signal E_b is 7 dB higher than n_{max} at –92 dBm. This value is less than $1pW$ of the received power or equivalently, with the mean voltage value being around 6 mV in a 50 Ω system.

On the other hand, the maximum input level test case is specified at –19 dB [132]. This amount is the ratio of $DPCH\_E_c$ to the total transmitted power spectral density (PSD) at the antenna port. Since the received PSD at the antenna connector of user equipment (UE) is –25 dBm, the $DPCH\_E_c$ is –44 dBm. Similar to minimum input level test case, where the G_p is 25 dB, the E_b is found to be –19 dBm. In this case, the mean voltage value is about 25 mV. Hence, the difference between the maximum and minimum levels is approximately 73 dB.

4.2.4 Adjacent Channel Selectivity

To avoid excessive signal mixing, any signal from an adjacent channel ($\pm$ 5 MHz from the wanted channel) that has higher power than the signal in the wanted channel must be suppressed. Therefore, the adjacent channel selectivity (ACS) is defined for the Rx and it must be greater than 33 dB [132]. According to ACS, the $DPCH\_E_c$ in the wanted channel is –103 dBm/3.84 MHz, which is 14 dB above the minimum sensitivity limit. Considering the 25 dB improvement from G_p, the bit energy E_b is –78 dBm. Meanwhile, the maximum PSD of the adjacent channel measured at the wanted channel's antenna connector must be –52 dBm per channel (Fig. 4.2). With ACS as 33 dB, the maximum interference PSD is

–85 dBm.

4.2.5 Blocking Characteristics

Blocking characteristics are categorised into in-band and out-of-band blockings. The in-band blocking is particularly important since it contains adjacent channel signals at 10 and 15 MHz away from the wanted channel. In this case, $DPCH\_E_c$ must be better than 3 dB, pushing its level to –114 dBm per channel. Here, E_b is at –89 dBm, and the maximum interference level is 7 dB down to –96 dBm. The in-band blocking is difficult to be suppressed completely due to imperfections of the filter. Since the channel is still close to the wanted channel, significant signal leakage could occur. For out-of-band blocking, in the presence of continuous wave (CW) interferers, the BER must not exceed $10\exp(-1)$ [30]. The ACS and the blocking specifications are illustrated in Fig. 4.2. The data in this figure are tabulated in Table 4.1.

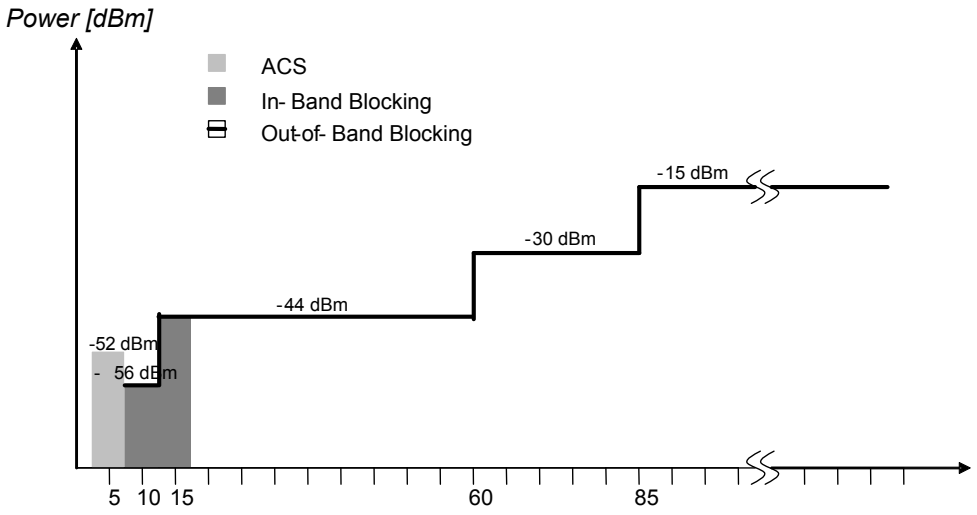


Figure 4.2: ACS and blocking characteristics for UMTS/FDD band I

Table 4.1: Parameters for the ACS and Blocking characteristics

Interferer	Power level and frequency
ACS	$-52dBm/3.84MHz$ at $\pm 5MHz$
In-Band Blocking	$-56dBm/3.84MHz$ at $\pm 10MHz$ away $-44dBm/3.84MHz$ at $\pm 15MHz$ away
Out-of-Band Blocking	$-44dBm/3.84MHz$ at $15-60MHz$ away $-30dBm/3.84MHz$ at $60-85MHz$ away $-15dBm/3.84MHz$ at $85MHz$ and above

4.3 Design Consideration of an ADC

The dynamic range provided by the down-conversion process determines the minimum number of bits required for the Rx to convert the analog signal into a digital form for baseband processing. This range is derived from the specification [132] that influences the analog front-end, such as the amplification, mixing, and filtering processes. Usually, an automatic gain control (AGC) consisting of a VGA in its control loop is used before the ADC. This AGC amplifies the previously filtered signal to reduce the required dynamic range requirement for the ADC and to adjust the DC level for the ADC input. In Zero-IF Rx, the channel select filter is used because the downconverted signal is in the baseband range.

The choice of ADC for UMTS applications is also of great interest for system designers. At the moment, two architectures are very popular. While the $\Delta\Sigma$ converter is used mainly for base station, the pipeline ADC is the choice for user-end terminal, also known as a mobile station (MS) [35]. A $\Delta\Sigma$ ADC is normally used with Low-IF or super-heterodyne architectures, as it is easier to control its performance. The pipeline type, on the other hand, is a choice for MS due to its small circuit complexity. Since the pipeline ADC also has granular attributes that can be used to scale power consumption, it is chosen to implement this design at the later stage of the thesis.

4.4 Derivation of ADC Specifications

4.4.1 ADC Dynamic Range and Resolution Estimation

The maximum detectable signal of DPCH signal is –44 dBm, as shown in Fig. 4.3. Meanwhile, the maximum cumulative noise and interferer signal are specified from the signal selectivity, which is –85 dBm. This amount has taken into account the ACS of 33 dB, SNR at the ADC input of –18 dB for modulated signal, and a noise distance of 14 dB up from the minimum reference sensitivity in the blocking test case.

The maximum gain provided by the AGC is typically around 50 dB [35,136] to improve the maximum detectable level around +6 dBm and minimum interference level to –35 dBm. Therefore, the dynamic range at the analog end is:

$$+\,6\,dBm - (-35\,dBm) = 41 \quad dB \tag{4.10}$$

The dynamic range of the ADC as defined in [137] is:

$$SNR_{ADC} = 6.02 \times N + 1.76 \quad dB \tag{4.11}$$

For simplicity, the SNR provided by each bit of the converter is 6 dB, then for a 41 dB dynamic range, ideally the required number of bits N is

$$41\,dB/6\,dB = 6.8 \quad bits \tag{4.12}$$

Practically, extra bits are required for the errors due to the offsets, the dynamic transient for settling effect, and the CF to ensure linearity. Thus, 10 $bits$ would be the desired N_{min} value for the ADC. If the maximum voltage for the ADC is considered to be 1 volts peak-to-peak (V_{pp}), which is +4 dBm, then a 2 dB margin is already taken into account from the AGC amplification.

In the case of software radio implementation, the analog front end of the Rx is ideally eliminated. The ADC is then placed directly after the antenna and pre-

Figure 4.3: UMTS/FDD signal test cases

select filter. Therefore, the calculation for the dynamic range is different. In this case, the maximum received power at the antenna input is –25 dBm (12.6 mV_{rms} in a 50Ω system, where V_{rms} is the root-mean-square voltage). Meanwhile, the minimum value is –106.7 dBm (1μV_{rms} in a 50Ω system), which is the minimum received PSD at the antenna input. Again, if the maximum input voltage for the ADC is 1 V_{pp}, then the required SNR will be:

$$SNR = 20\log_{10}(1\,V_{pp}/(2\times\sqrt{2}\times 1\,\mu V_{rms})) = 110.97 \quad dB. \tag{4.13}$$

Another 6 dB is considered for signal decoding and 20 dB for the offsets, the dynamic transients and the CF, then the dynamic range of 137 dB is required. This is equivalent to using an ADC of:

$$137\,dB/6\,dB \approx 23\,bits \tag{4.14}$$

This condition is difficult to achieve using current technology. Furthermore, since the ADC is placed immediately after the antenna and within the transmitted radio frequency, a very high f_s reaching up to tens of GHz is required. This is another limitation in current ADC design. Hence, using an analog front-end to shift the radio frequency signal down to the MHz range would be a good choice.

4.4.2 ADC Sampling Frequency for $I-Q$ Zero-IF Receiver

The bit rate of 12.2 kbps of a voice signal is coded and spread into a R_c of 3.84 Mcps equivalent to a 3.84 MHz signal bandwidth containing information and 99.9% of the energy concentrated from 5 MHz bandwidth. This signal consists of modulated I and Q components in a time-interleaved format. At the Rx, the signal is split into two separate I and Q components after the down-conversion process. Each component carries 1.92 Mcps of data bits and control bits, respectively. The data signal ω_m of 3.84 MHz is transmitted using a carrier angular frequency ω_c of around 2 GHz. This can be represented with:

$$s(t) = \cos([\omega_c - \omega_m]t) \tag{4.15}$$

where $s(t)$ denotes the signal sent at the Tx part. To simplify the calculation, we assume unity amplitude, ideal transceiver behaviour, and ideal channel conditions. At the Rx part, the received signal $r(t)$ after the LNA is downconverted to the baseband frequency by multiplying $r(t)$ with the carrier frequency ω_c in both I and Q paths.

$$\begin{aligned} i(t) &= \cos([\omega_c - \omega_m]t).\cos(\omega_c t) \\ &\quad + \dots \\ &= 0.5\ \cos([2\omega_c - \omega_m]t) + 0.5\ \cos(-\omega_m t) \\ &= 0.5\ \cos([2\omega_c - \omega_m]t) + 0.5\ \cos(\omega_m t) \end{aligned} \tag{4.16}$$

$$\begin{aligned} q(t) &= \cos([\omega_c - \omega_m]t).\sin(\omega_c t) \\ &\quad + \dots \\ &= 0.5\ \sin([2\omega_c - \omega_m]t) - 0.5\ \sin(-\omega_m t) \\ &= 0.5\ \sin([2\omega_c - \omega_m]t) + 0.5\ \sin(\omega_m t) \end{aligned} \tag{4.17}$$

Since the frequency around $2\omega_c$ is high enough and can be filtered out, the components to the ADC inputs consist of ω_m.

$$i(t) = \cos(\omega_m t) \tag{4.18}$$

$$q(t) = \sin(\omega_m t) \tag{4.19}$$

If $5MHz$ for ω_m is considered, the required $f_s(min)$ for a Nyquist ADC is approximately $10MHz$.

4.5 Summary

A design consideration for a Nyquist-rate ADC to be used in future mobile communication Rx is presented. A mathematical analysis method is used to show that the ADC requirements are related to the Rx specifications from UMTS governing documentations. As an application platform, a direct-conversion Rx architecture topology, suitable for mobile phones, is considered. The UMTS/FDD standard specifications and documentations from 3GPP are used to specify the minimum required N and f_s for the ADC. Since the architecture utilises both I and Q channels, it is concluded that N_{min} is 10 $bits$ and $f_s(min)$ is $10MHz$ for a practical implementation. The required resolution takes into account additional bits to counter the analog design's imperfections.

The next chapter proposes an adaptive control scheme for ReADC that is suitable for low-power devices. The algorithm is signal-dependent, and the ReADC adjusts its N adaptively according to the amplitude variation of the signal.

Chapter 5

Adaptive Algorithm for ReADCs

Contents

5.1 Introduction

In electronics, an adaptive system is capable of changing its internal behaviour to suit a desired operation based upon the available resources. Often, this is achieved by evaluating the performance of the current operation to the pre-determined reference values. This is similar to the feedback control system that is ideally used to regulate the input to achieve the desired outputs. The reference values are either fixed throughout the operation or adjusted at each observed interval.

However, adjustment of these values is still bounded to the specified conditions or a set of rules.

The operation of an adaptive system is determined by its associated algorithm. Usually, the algorithm is developed based upon observational study of natural behaviour and its conventional operation. The algorithm acts as a control procedure to monitor the changes continuously and adaptively, and to compute its suitable response to the changes. Generally, a control procedure involves a monitoring process of measuring, evaluating, comparing, and adjusting.

This chapter presents a new adaptive technique for ReADC architectures. The technique exploits the switching activity (SA) nature of the output of the ADC. The digital outputs of ADC are monitored, evaluated, and compared continuously to its reference values to trigger the configuration signal adaptively. A pipeline ReADC is used to demonstrate the functionality of the algorithm. The adaptive scenario considers an implementation of ReADC for reN.

Firstly, the UMTS signal is generated using the MATLAB function. This is achieved by thoroughly studying the UMTS documentations and simulating the building blocks for the UMTS/FDD receiver. The generated signal is then analysed in terms of stationarity and periodicity by considering its framing structure. Secondly, a method using the SA nature of the ADC output is proposed. Consequently, the method is used as the fundamental of the adaptive algorithm for ReADCs. Several reconfigurability issues regarding the algorithm are also discussed. Then, the algorithm is implemented in the pipeline ReADC to observe its functionality. Finally, the BER performance is evaluated.

5.2 Generation of UMTS Signal

A detailed description of UMTS signalling as described in [29, 30, 132, 138–140] is studied carefully. Specifically, the physical layer of the UMTS signal and the RF characteristics are reviewed thoroughly.

The scope of the thesis is limited to the downlink path, a link between a base station's transmitter and a mobile station's receiver. Therefore, only UMTS signalling for this downlink is considered. The signal for the downlink has different specification than the uplink path, especially in terms of its framing and RF limitations.

5.2.1 UMTS Downlink Building Block

Figure 5.1 illustrates the building block of the UMTS digital modulator for the downlink. The information data for the downlink carried by the dedicated physical data channel (DPDCH) are multiplexed with the control data carried by the dedicated physical control channel (DPCCH). However, the modulation process is separated between the DPDCH assigned to I-channel and the DPCCH carried by Q-channel. Therefore, the data first need to be split from serial to parallel. Both I and Q paths consist of spreading process and a QPSK mapper. The spreading process involves two processes: channelisation and scrambling.

Channelisation is a process where the data is spread with an orthogonal variable spreading factor (OVSF) with different values of a spreading factor (SF). The SF is the ratio of the chip rate R_c to the data bit rate R_b. For the I-channel, the value of SF varies from 4 to 512. To achieve the best protection against interference, the SF value for the Q-channel is fixed to 256. Then, both channels are combined into a complex data. After channelisation, the complex valued data undergo a scrambling process, wherein the data is multiplied with a scrambling code that is repeated for every frame [29, 30]. The complex value scrambling codes are generated from a real Gold code [30]. Later, the data is split into I- and Q-channels for the pulse-shaping process using RRC filters.

The RF part of the downlink is a transmitter consisting of DACs, reconstruction filters, mixers, a local oscillator, a combiner, a power amplifier (PA), and an antenna. These blocks are analog circuits. The UMTS digital modulator is connected directly to this RF unit. Figure 5.2 illustrates the building blocks for

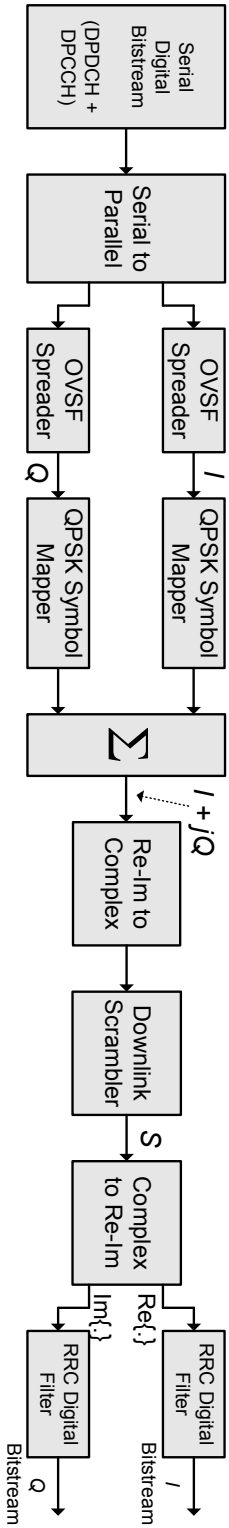


Figure 5.1: UMTS digital modulator

the UMTS direct conversion transmitter. Usually, an upsampling process followed by a digital interpolation filter take place before the DAC to match the sampling frequency between baseband and the RF. Upsampling is used to relax the requirement of the filter. Furthermore, baseband frequency is normally set to the minimum possible value to reduce the power dissipation of the digital circuits.

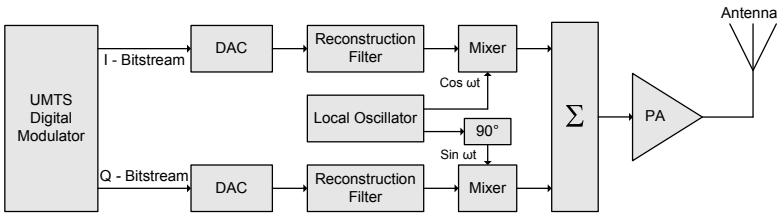


Figure 5.2: UMTS direct conversion transmitter

5.2.2 UMTS Signal

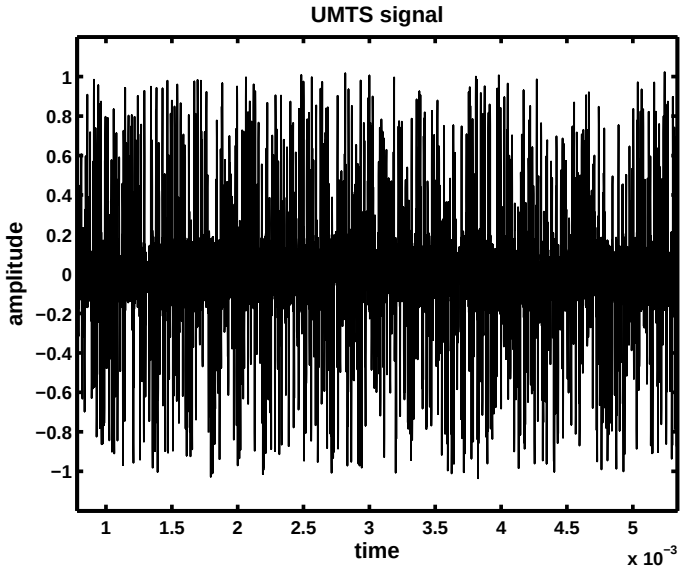


Figure 5.3: UMTS signal

The behaviour of the building blocks of Fig. 5.1 and Fig. 5.2 is described in MATLAB codes. For simplicity, the PA is represented by a gain and the antenna

is neglected. Moreover, a noiseless condition is assumed.

Figure 5.3 shows the UMTS signal generated using MATLAB. The amplitude of the signal varies between $+1V$ and $-1V$ and the frequency is 3.84 MHz. The plot shows the signal of approximately half of the frame duration.

5.3 Analysis of UMTS Signal

The data of UMTS frame is constructed into a number of slots, and each slot always contains 2560 chips. Since the frequency of R_c is 3.84 MHz, each chip duration T_{chip} is 0.26042μs. Hence, a total slot duration T_{slot} of 2560 chips is 0.6667ms.

A UMTS frame, which consists of 15 time slots grouped together, has a frame duration T_{frame} of 10ms. Meanwhile, a total of 72 frames is known as a superframe $T_{superframe}$ with a duration of 720ms, as shown in Fig. 5.4.

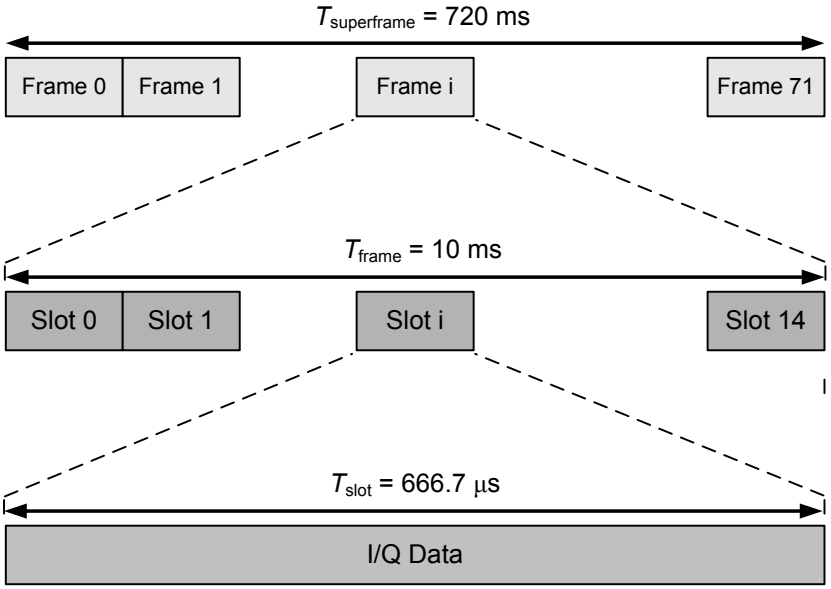


Figure 5.4: UMTS framing

The same scrambling code is repeated for each frame of 10ms duration. Therefore, the UMTS signal can be viewed as having cyclostationary properties, which

have multiple interleaved stationary properties. These attributes were used by [141, 142] to develop a sensing technique in UMTS. An analytical formulation of a UMTS signal with cyclic autocorrelation properties for a UMTS/FDD signal is explained and analysed extensively in [143, 144]. Furthermore, a frame length of $10ms$ is equivalent to the shortest possible transmission time interval (TTI), the duration of a transmission on the radio link [29]. Similarly, the shortest SMS is sent over a frame length [30]. Since a UMTS frame length is considered to have many attributes mentioned above, a duration of $10ms$ is a minimum window to observe the UMTS signal periodically.

5.4 Switching Activity Measurement (SWAM) Method

In this section, the SA concept used as the methodology is defined. It is based on the hamming distance (bit transition) between the successive digital outputs. In addition, the operation of the proposed solution is presented.

5.4.1 Observation

ADC is the most common module in modern electronic devices [145]. Since it is widely used, apart from the normal operation of converting the analog signal into digital form, additional functionality that can be deduced from the device will be extremely beneficial. Inspired by the negative feedback control amplifier concept, invented by Harold Stephen Black at Bell Laboratories in 1927 [146], the digital output of an ADC has to be manipulated to control a system. The advantage of this approach is that because the ADC output is in the digital form, manipulation will be easier. In this case, this does not require typical controller types, similar to the normal control theory as described in [147]. The controller is a kind of digital module, and the output of the controller is used to trigger configuration switches. To achieve the goal of using the ADC as a control mechanism, the behaviour of the

ADC output (i.e., digital data) must be investigated first to extract any relevant, useful information.

5.4.2 Nature of ADC Output

An example of an analog signal and its equivalent SH outputs at each sampling interval, t_s is illustrated in Fig. 5.5. Digital equivalent (6-*bit* ADC outputs) of the first 16 samples are tabulated in Table 5.1. In this example, t_s is $12.5ms$. This is equivalent to f_s of $1/t_s$, which is $80Hz$. The full-scale range of a 6-*bit* ADC is $2V$. This defines the LSB voltage V_{LSB} of $2/2^6V$, which is $31.25mV$. A large variation of sampled data is denoted by marker '▽' whereas a marker '**O**' indicates a small variation of sampled data . These data are the SH outputs of an analog signal, as indicated by the black curve.

As per Table 5.1, large variations of sampled data (samples 1 to 4 and samples 9 to 12) lead to a significant variation of their equivalent ADC outputs. In contrast, small variations of the sampled data (samples 4 to 8 and samples 13 to 16) have similar (or almost similar) ADC outputs.

This observation concludes that if an analog signal is highly fluctuated, so too is its equivalent ADC output. Meanwhile, if the analog signal slowly varies over time (low frequency), then the ADC output also shows low variation. Consequently, this observation enables this information to be used as a principle method to monitor and control a ReADC adaptively.

5.4.3 Hamming Distance

5.4.3.1 Overview

The SA is defined as digital bit transition either from zero to one ($0 \rightarrow 1$) or one to zero ($1 \rightarrow 0$) [148, 149]. In the digital design, the SA contributes to the dynamic power consumption P_{dyn} given by [150, 151] as

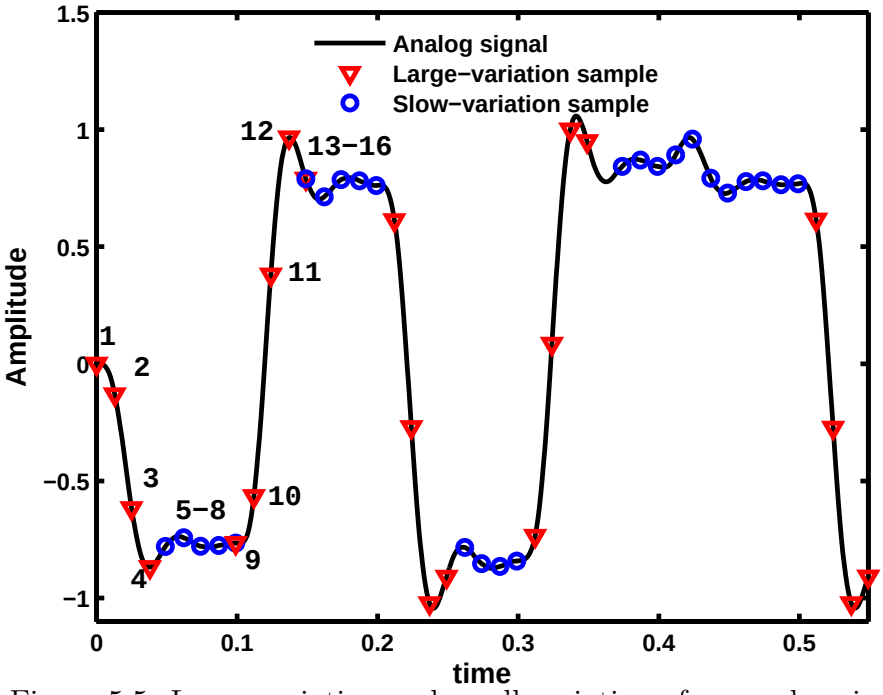


Figure 5.5: Large variation and small variation of an analog signal

Table 5.1: Digital output of ADC of Fig. 5.5

sample	time $[ms]$	$\mathbf{V_{sample-hold}}$[V]	ADC digital output
1	0	0	100000
2	12.5	−0.150	000101
3	25	−0.600	010011
4	37.5	−0.900	011101
5	50	−0.850	011011
6	62.5	−0.800	011010
7	75	−0.820	011010
8	87.5	−0.810	011010
9	100	−0.815	011010
10	112.5	−0.550	010010
11	125	+0.400	101101
12	137.5	+0.950	111110
13	150	+0.750	111000
14	162.5	+0.650	110101
15	175	+0.745	111000
16	187.5	+0.745	111000

$$P_{dyn} = \sum_{i} V_{dd}.V_{swing}.C_{Load}.\alpha_{0\to 0}.f \tag{5.1}$$

where V_{swing} is full-range input voltage, C_{Load} is load capacitance, f is clock frequency, and $\alpha_{0\to 1}$ is SA at nodes. Since the SA that occurs at nodes is difficult to evaluate, estimation models are normally used [150, 152–155]. To reduce the activity, careful design methods at gates level are introduced [148, 149].

5.4.3.2 Hamming Distance Concept

The hamming distance concept was introduced originally for the detection and correction of errors in digital communication [156]. It defines the number of bit changes between two digital vector or bit strings. In other words, it represents the relative distance of a vector from another. As illustrated in the Fig. 5.6, the hamming distance of two bit strings, A and B is calculated by summing total number of 1s from the result of XOR-ed A and B. In this example, the hamming distance is 5.

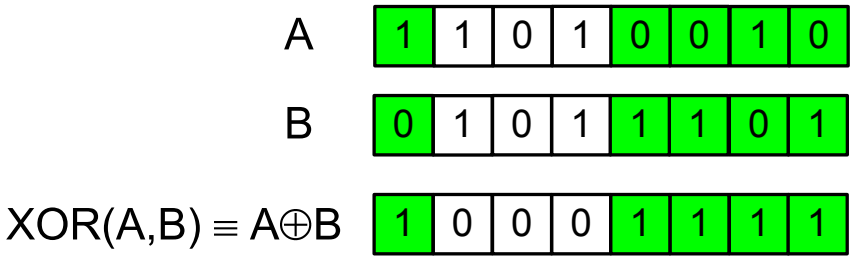


Figure 5.6: Hamming distance of two digital bit strings

If the above strings, A and B represent the ADC outputs of two adjacent samples of analog signal, then the hamming distance measures the number of bit changes between the transition of a sampled signal to the next sample. The hamming distance concept is used to evaluate the total SA value, S_t of the accumulated samples. Using this approach, measuring signal changes in digital form is

easier compared with the original analog form, because the ADC already digitises the signal.

5.4.4 Binary-Weighted Hamming Distance

Measuring changes using the traditional hamming distance approach is not suitable for the ADC wordlength because each bit has a different weighting. For example the MSB bit of a N-*bit* wordlength has a weighting of 2^{N-1} instead of the LSB bit with a weighting of 1. Furthermore, normal hamming distance computation only counts the bit changes without considering the position of each bit. For the case of ADC wordlength, each bit position is important. Instead, to improve the SA computation, binary-weighted hamming distance is used.

Consider a result of XOR output between two ADC wordlengths as shown in Fig. 5.7. This result is multiplied with a binary coefficient β. Then, the multiplication result of each bit is accumulated. In the figure, using the binary-weighted hamming distance resulted in a SA of 143 compared with 5 using the conventional method. This has improved the degree of freedom (DoF) to distinguish between results. In this case, each bit position has been considered with an appropriate weighting associated with the amplitude of the analog signal.

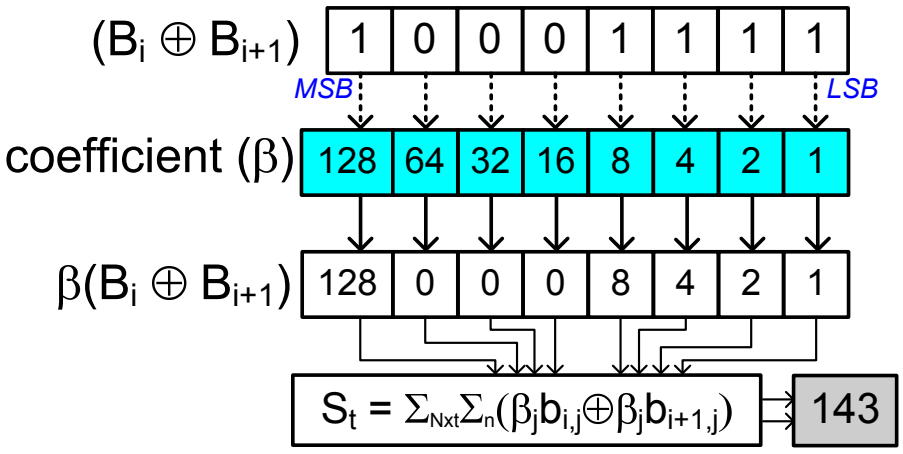


Figure 5.7: Binary-weighted hamming distance of two digital bit strings

5.5 A Proposed Adaptive Algorithm using SWAM Method

In this section, an adaptive algorithm for ReADC is proposed. This method is based on the following observations and principles:

1. The ADC output represents a digital equivalent of a sampled analog signal.
2. A large variation in analog signal leads to a large variation in ADC output and vice versa.
3. The binary-weighted hamming distance can be used to measure the bit changes between two consecutive ADC outputs.
4. A large value of hamming distance indicates that the analog signal is heavily fluctuated, and vice versa.
5. A large variation of analog signal requires a higher resolution to cover the full-scale signal range.
6. Binary-weighted coefficients are used to consider each bit position of an ADC wordlength.

The algorithm based on the proposed SWAM method is presented in Algorithm 1. Initially, the controller is set with the threshold (reference) values. Subsequently, a number of samples, n in a measurement interval $T_{measure}$ is defined. This allows an accumulative SA value to be measured periodically during a specific time frame. A total duration for an observed cycle T_{win} is defined. This duration specifies a measurement interval $T_{measure}$ and a period T_{idle}, in which no measurement is made. During this operation interval, the algorithm is idle after the measurement is made. At the end of T_{win}, a new measurement is made before deciding on a new change. During the measurement period of $T_{measure}$, the

Algorithm 1 Adaptive pseudo-code

1: Initialise parameters; set threshold values vt_l; set number of samples n in a measurement interval $T_{measure}$; set an observed window duration T_{win} for each conversion cycle
2: Begin conversion with N-bit resolution
3: Measure difference in SA, α_i of two successive ADC digital outputs, B_i and B_{i+1} within an interval of $T_{measure}$ of n samples
4: Accumulate the total SA of digital outputs S_t within this interval $T_{measure}$
5: Compare S_t at the end of each interval $T_{measure}$ with the threshold values vt_l
6: Decide on changes (N_{new})
7: Change the converter resolution to this N_{new}
8: Continue conversion with N_{new} for the rest of observed window, T_{idle} ($T_{win} = T_{measure} + T_{idle}$)
9: Repeat step 3 at the end of T_{win}

converter operates with the previous computed N value. In short, the converter cyclically changes its resolution at the end of $T_{measure}$.

Similarly, threshold values vt_l are set depending on different levels of N that is chosen. For example, if the maximum resolution N_{max} is 10 and the minimum resolution N_{min} is 6, there are five different N values. In this case, four threshold values are required to distinguish the SA result between levels. Therefore, the number of level l is computed as (5.2).

$$l = N - 1 \tag{5.2}$$

The threshold values set the limit of maximum and minimum allowable SA for the converter to operate with the current N value. A detailed discussion on parameter settings will be presented in the next chapter.

With the preloaded references, the converter begins its operation with *N-bit* resolution and sampling-rate f_s. Normally for adaptive operation, either one of these performance parameters is fixed. However, both parameters can be simultaneously set as variables. Nevertheless, this is not within the scope of the thesis.

In an interval $T_{measure}$ of n samples, digital data (ADC output) at the i-th sample, B_i is compared with the previous sample, B_{i-1}, that corresponds to the $(i-1)$-th sample. In this case, B_1 is the first sample, B_n is the last sample, and

β is the binary-weighted coefficient associated to each bit position. During this interval, intermediate SA, α_i is measured between two successive digital outputs, as presented by (5.3).

$$\begin{aligned} \alpha_i &= \beta \ |B_i - B_{i-1}| \\ &= \beta \ XOR(B_i \, , \, B_{i-1}) \\ &= \beta \ (B_i \oplus B_{i-1}) \end{aligned} \tag{5.3}$$

In hardware implementation, this is done simply using XOR logic. Altogether, there will be $(n-1)$ differences in the digital output corresponding to α_{n-1} intermediate SAs in this interval. α_i is accumulated for the whole interval of $T_{measure}$. At the end of $T_{measure}$, the total SA sum is computed and is represented as S_t in (5.4).

$$\begin{aligned} S_t &= \beta \sum_{i=1}^{n-1} \alpha_i \\ &= \beta \sum_{i=1}^{n-1} XOR(B_i \, , \, B_{i-1}) \\ &= \beta \sum_{i=1}^{n-1} B_i \oplus B_{i-1} \end{aligned} \tag{5.4}$$

or equivalent to

$$\begin{aligned} S_t &= \sum_{j=1}^{N} \sum_{i=1}^{n-1} \beta_j b_{i,j} \oplus \beta_j b_{i+1,j} \\ &= \sum_{j=1}^{N} \sum_{i=1}^{n-1} 2^{j-1} (b_{i,j} \oplus b_{i+1,j}) \end{aligned} \tag{5.5}$$

where $B = [b_N \ b_{N-1} \ b_{N-2} \ ... \ b_2 \ b_1]$, and b_j is associated to the bit position in an ADC wordlength according to N. Since binary weighting is used, the coefficient b_j corresponded to 2^{j-1} at a particular location. Here, b_N and β_N are the MSBs, whereas b_1 and β_1 are the LSBs. Equations (5.4) and (5.5) can also be written as

$$
\begin{aligned}
S_t &= \beta \sum_{i=2}^{n} XOR(B_i\,,\, B_{i+1}) \\
&= \beta \sum_{i=2}^{n} B_i \oplus B_{i+1} \\
&= \sum_{j=1}^{N} \sum_{i=2}^{n} \beta_j b_{i,j} \oplus \beta_j b_{i+1,j} \\
&= \sum_{j=1}^{N} \sum_{i=2}^{n} 2^{j-1}(b_{i,j} \oplus b_{i+1,j})
\end{aligned}
\tag{5.6}
$$

At the end of each period of $T_{measure}$, S_t is compared with the threshold values vt_l. The controller will decide to increase, decrease, or maintain the current resolution of the ADC.

If S_t is larger than vt_l, then the control flag is triggered to increase N to a new N value, N_{new}. Once the flag is set, a control pulse triggers the converter to change its resolution from N to N_{new} and continue the conversion with this new resolution for the interval T_{idle}. Otherwise, if S_t is smaller than vt_l, then N is maintained at the current value or is reduced to a lower resolution set by the appropriate threshold value. The resolution remains unchanged for the next conversion interval T_{idle}.

At the system level application, to avoid large number of computations, S_t value can be normalised as a fraction. Usually, the denominator is the maximum possible value that can be computed from a reference signal. Consider the digital output of a 6-*bit* ADC of five samples per $T_{measure}$ for two successive measurement interval T_i and T_{i+1}, as in Table 5.2. Note that the interval T is equivalent to $T_{measure}$ from the previous description. Then, α_1 and α_2 between B_1, B_2, and B_3 for T_i interval are computed as in Table 5.3.

The output changes between successive outputs α_1 (i.e., XOR(B_2,B_1)) and α_2 (i.e., XOR(B_3,B_2)), are calculated to be 5 in both cases. In total, S_t is 17 within this interval T_i and 85 for the interval T_{i+1}. Theoretically, if the digital output

Table 5.2: Digital output of 6-*bit* ADC for two adjacent intervals

ADC Output	$\mathbf{T_i}$	$\mathbf{T_{i+1}}$	$\mathbf{T_{max\ changes}}$
B_1	000011	000100	000000
B_2	000110	000110	111111
B_3	000011	000101	000000
B_4	000011	010100	111111
B_5	000100	101011	000000

Table 5.3: SA computation between two samples

B_2:	000110	B_3:	000011
B_1:	000011	B_2:	000110
$(B_2 \oplus B_1)$:	000101	$(B_3 \oplus B_2)$:	000101
β:	000401	$(B_3 \oplus B_2)$:	000401
$\beta(B_2 \oplus B_1)$:	5	$\beta(B_3 \oplus B_2)$:	5

changes drastically from zero (000000) to a FS range (111111) at each sample, as shown in interval $T_{max\ changes}$, then the activity will be maximum $S_t(MAX)$. This can be represented as (5.7):

$$S_t(MAX) = (n-1).(2^N - 1) \tag{5.7}$$

For example, in this interval, $S_t(MAX)$ is computed as 252. In each period T, the normalised value of S_t will be

$$S_t(norm) = \frac{S_t}{S_t(MAX)} \tag{5.8}$$

In reality, $S_t(MAX)$ is very unlikely. However, this value is used for a reference during the computation.

Consider the framing structure as illustrated in Fig. 5.8 for the simulation of the entire signal. If there is a total of q frames available, then the entire signal duration is $10q\ ms$. Therefore, for the total number of q frames (each frame measures n samples of data), the mean value of S_t, $\langle S_t \rangle$ of the total sample is

computed by (5.9) as

$$\langle S_t \rangle = \frac{\sum_{k=1}^{q} S_{t_k}}{q} \tag{5.9}$$

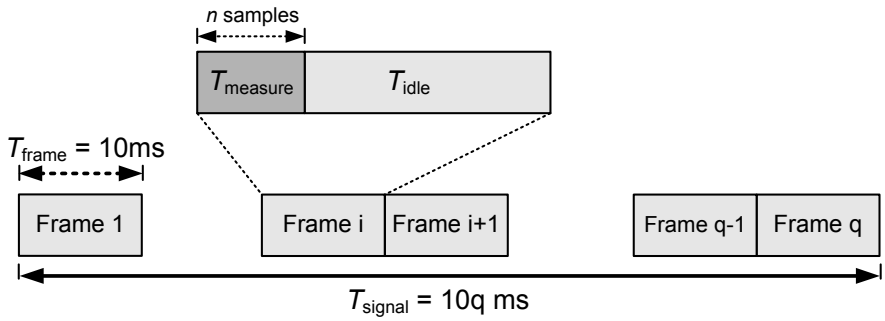


Figure 5.8: SA computation for the whole UMTS sample signal

5.6 Design Issues

Four issues need to be considered. First, the SA computation depends on ADC wordlength or N. As N increases, the arithmetic computation also increases, and hence, S_t will grow larger. Therefore, to avoid an ever-increasing S_t value, an approach to minimise the computational complexity is discussed. Secondly, the SA is measured within a period of $T_{measure}$. This corresponds to n number of samples for the computation. As a result, the determination of the n value is presented. The third issue is the setting of threshold values to ensure correct operation. Hence, the correlation between SA and different levels of signal amplitude is analysed initially and the threshold values are derived from this analysis. Finally, since ReADC involves a change in ADC wordlength, an issue of data transfer between the ReADC module and the baseband is also discussed. Since the ADC wordlength is changed continuously throughout the operation, the next processing block must be aware of this and a solution is presented for this issue. The following subsections will discuss these issues in detail.

5.6.1 Number of Bits for Switching Activity (SA) Computation

At the end of Section 5.5, an example of a wordlength of 6-*bit* ADC is discussed. Now, suppose the resolution of the ADC is increased to 16 *bits*, where the first five digital data of the sampled signal are tabulated in Table 5.4, the effect of different lengths is observed. Note that, the first six MSBs are equal to the lists in Table 5.2.

Table 5.4: Digital output of a 16-*bit* ADC for two adjacent intervals

ADC Output	$\mathbf{T_i}$	$\mathbf{T_{i+1}}$
B_1	0000110101011111	0001001000111000
B_2	0001101110100001	0001100001110101
B_3	0000110001111001	0001011001100110
B_4	0000111100011000	0101000000111101
B_5	0001000000111000	1010111100000110

Using a 16-*bit* wordlength, the intermediate values α_1 and α_2 are calculated to be 5886 and 6104, respectively. In an interval T_i, $S_{t(i)}$ is 20,823 and $S_{t(i+1)}$ is found to be 89,590 in the interval T_{i+1}. As compared with the previous 6-*bit* wordlength, these values have increased significantly for both intervals. If all N-*bit* resolutions are used for SA computation, S_t grows larger as N increases.

To decrease the calculation complexity, S_t is measured for the first N-effective-MSB *bits* of the SA calculation, N_{xt}. Furthermore, by setting a fixed N_{xt} value, only a coarse variation step is considered. Fine step changes can be ignored. For example, consider 12-*bit* ADC with FS input range of $1V$. Thus, the V_{LSB} is equal to $1/2^{12}$ V, which is $244\mu V$. If N_{xt} is 6 *bits*, then the converter only measures any changes larger than $1/2^6$ V, which is approximately $16mV$. This approach can be considered if the converter varies its N with coarse step changes in amplitude. This is particularly useful for low frequency applications since the amplitude varies slowly and the peak occurs at a very low rate.

However, due to the noise-like nature of the wireless signal that changes quickly over time, each bit of the ADC output is very important to distinguish the amplitude variation. In other words, very fine step changes can only be determined by considering each bit of the ADC wordlength. Therefore, the design in this thesis only considers the whole length of the 10-*bit* resolution.

A sample of the UMTS signal is used in the simulation, where the N value for the SA computation, N_{xt} is changed from 1-*bit* to 10-*bit*. The result of the MATLAB simulation for various UMTS slots is shown in Fig. 5.9.

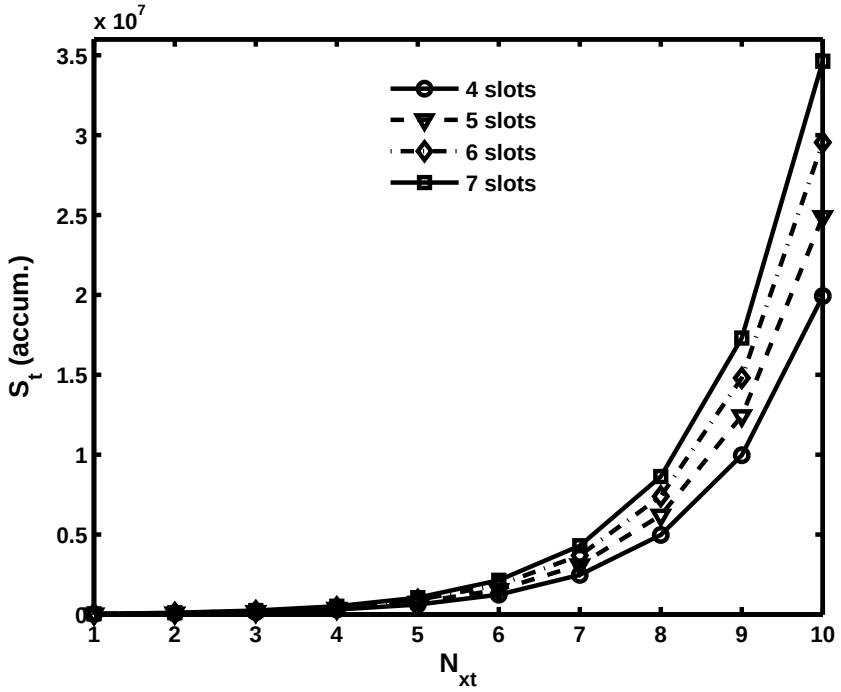


Figure 5.9: Variation of SA results for four to seven time slots

From Fig. 5.9, the result verifies that for the SA computation, the accumulative SA, $S_t(accum.)$ increases proportionally with N_{xt}. Similarly, with the increment of the time slot that denotes more samples is used, SA also increases. Moreover, since binary weighting is used, the result shows logarithmic curves. To simplify the result, by considering N_{xt}, S_t in (5.5) is computed as

$$\begin{aligned} S_t &= \sum_{j=1}^{N_{xt}} \sum_{i=1}^{n-1} \beta_j b_{i,j} \oplus \beta_j b_{i+1,j} \\ &= \sum_{j=1}^{N_{xt}} \sum_{i=1}^{n-1} 2^{j-1}(b_{i,j} \oplus b_{i+1,j}) \end{aligned} \tag{5.10}$$

Similarly, the theoretical maximum SA, $S_t(MAX)$ in (5.7) is equal to

$$S_t(MAX) = (n-1).(2^{N_{xt}} - 1) \tag{5.11}$$

5.6.2 The Length of the Measurement Period ($T_{measure}$)

In the design, the sampling frequency of the ADC is four times faster than the signal frequency of 15.36 MHz (or $15,360,000$ samples per second). Therefore, in a frame duration of $10ms$, there will be 153,600 samples. This is equivalent to 10,240 samples for each slot of $666.67\mu s$ duration. To determine the period of $T_{measure}$, UMTS framing is considered since each slot has a specific time window.

Consider the framing structure in Fig. 5.4. In this case, the duration for $T_{measure}$ and T_{idle} is considered to have a 50:50 ratio. Since the total slots in a frame are 15, the length of $T_{measure}$ is set to a half of the frame length. In this case, the measurement will be made during the first seven slots before the algorithm is set to the idle state (no computation) for the last eight slots after the result of SA computation is obtained and the decision is made.

The result of $S_t(accum.)$ for seven UMTS slots is given in Fig. 5.10. In this case, a conventional 10-*bit* ADC is used, and its output is used by the SWAM module to calculate SA changes. For each frame, the result is well beyond the 30 million changes within a $T_{measure}$ duration of $4.7ms$. Altogether, it shows the computation result for each frame for more than 100 frames.

In seven slots, as many as 71,680 samples are used. Theoretically, the maximum available change given by (5.7) is approximately 70 million changes. There-

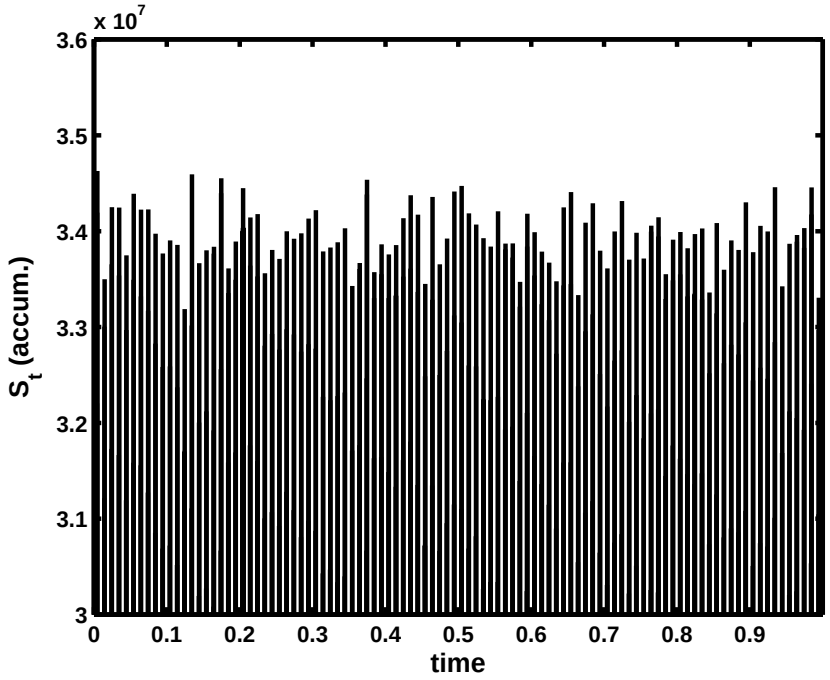


Figure 5.10: SA result of 10-*bit* ADC for seven time slots

fore, the result reveals that approximately less than half of the maximum value is computed during the simulation.

5.6.3 Threshold Values Setting

The threshold values are set by analysing the SA of the conventional ADC. In this study, a 10-*bit* ADC is considered. For the SA calculation, the entire ADC wordlength is used ($N_{xt} = 10$). In an interval of $T_{measure}$, $71,680$ samples are accumulated ($n = 71,680$). This is equivalent to seven UMTS slots. The amplitude is scaled down from 100% to 10% and the SA value is observed. Figure 5.11 shows the result.

By scaling down the amplitude, the range of the normalised SA of a UMTS signal is between 0.442 and 0.466. In this case, when the amplitude is reduced to 10%, SA is reduced by 5%. Since the maximum value of $S_t(accum.)$ is 34.6 million changes, a 5% reduction in SA value is equivalent to a 1.6 million reduction in activity. This information can be used to change the resolution of an ADC

adaptively according to amplitude variation.

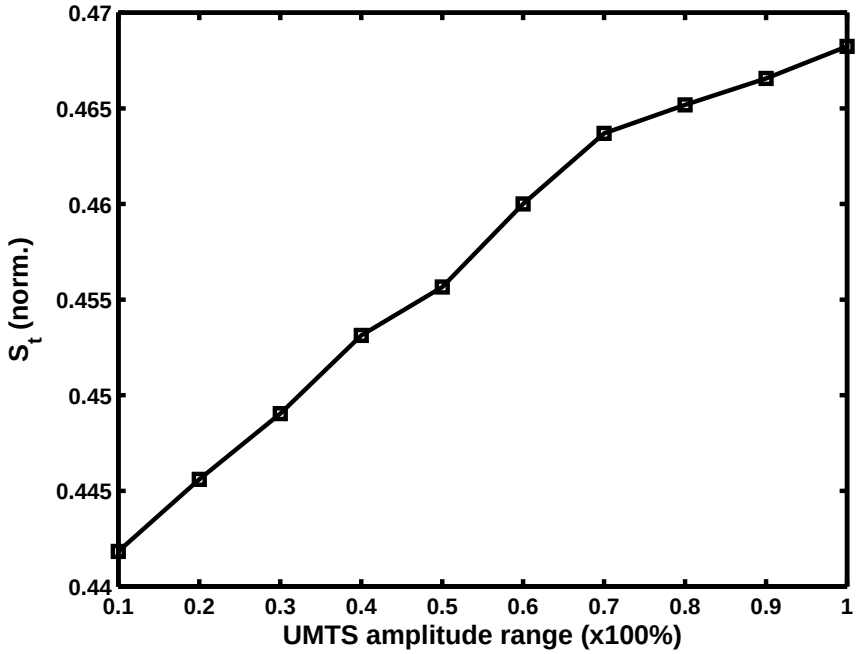


Figure 5.11: SA result against different UMTS signal amplitude for seven time slots

In this study, the threshold values are set by considering the half power reduction of the signal. In this case, a $-3dB$ drop in power is equivalent to an approximate 30% drop in voltage amplitude. This is equivalent of around 1% variation of the average SA, $S_t(avg)$ from the maximum-recorded SA value, $S_t(max)$ in Fig. 5.11. The $S_t(max)$ for seven slots is 0.468. Table 5.5 specifies the threshold values for this observation range. The highest threshold is 0.467, which is equivalent to 95% of the analog amplitude. The lowest threshold is 0.463, which is equivalent to 75% of the analog amplitude. The threshold values are set to be in this 1% margin or equivalently, within ±0.5% of the mid-point value. Thus, the adaptive algorithm responds with up to a 30% drop in magnitude.

The threshold value vt_1 sets the boundary between the 9 *bits* and the 10 *bits* operation. If S_t is greater or equal than vt_1, then the ADC will operate with 10 *bits* during the next period of T_{idle}. In contrast, if S_t is lower than vt_1, then N

Table 5.5: Threshold values setting for the adaptive algorithm

threshold (vt)	**value**
vt_1	0.467
vt_2	0.466
vt_3	0.464
vt_4	0.463

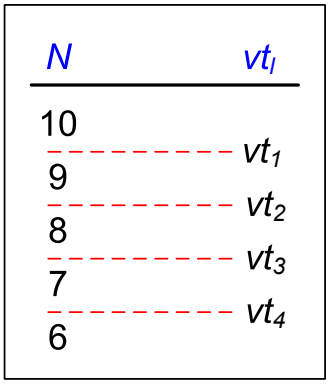


Figure 5.12: Threshold values setting for different N levels

will be assigned to 9 *bits* for the duration of T_{idle}. Similarly, vt_2 set the border between 9 *bits* and 8 *bits*, and vt_3 will distinguish between 8 *bits* and 7 *bits*. Likewise, vt_4 is the threshold between 7 *bits* and 6 *bits*. Therefore, 10 *bits* is the maximum resolution N_{max} and 6 *bits* is the minimum resolution N_{min}. Figure 5.12 summarises the above description.

The second approach is to set the value using a simple mathematical approach of the previous result. Theoretically, using (5.7), the theoretical maximum SA value, $S_t(MAX)$ for seven slots that consists of 10,240 samples using a 10-*bit* ADC is approximately 73.33 million changes. Referring to Fig. 5.10, the maximum measured SA value, $S_t(max)$ is around 34.6 million and the minimum measured value $S_t(min)$ is nearly 33.2 million. Thus, the normalised values are 0.472 and 0.454, respectively. The middle point between these two values is 0.463. From the statistical analysis represented by (5.9), $\langle S_t \rangle$ is also equal to 0.463 for the above example.

According to this result, $vt(max)$ is set to a middle value between $\langle S_t \rangle$ and $S_t(max)$. The $S_t(max)$ value is not chosen as $vt(max)$ since it has rare occurrence as can be seen from Fig. 5.10. Meanwhile, for the $vt(min)$ value, a middle value between $S_t(max)$ and $S_t(min)$ is chosen, which is 0.463. Since the $S_t(min)$ value is too small, by setting $vt(min)$ to this value, the adaptive parameter for reN is unlikely to operate at the minimum since S_t has exceeded $S_t(min)$ almost for each interval $T_{measure}$. The threshold value range is specified by these $vt(max)$ and $vt(min)$. Any intermediate value is chosen within this range according to different N values. For example, if N varies from 6 *bits* to 10 *bits*, then four different threshold levels are required. Since the values of $vt(max)$ and $vt(min)$ are similar, as discussed using SA and amplitude correlation; therefore, the threshold value setting using this method is similar to Table 5.5. For clarification, Fig. 5.13 simplifies the above description.

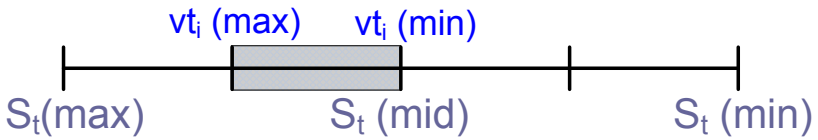


Figure 5.13: Mathematical approach for vt_l setting

5.6.4 Encoding Technique

In any ADC-based system, the output of ADC is passed to the next operational block. In the Zero-IF receiver implementation in Fig. 2.9, the ADC is connected to the baseband processor. In this case, the reconstruction filter that shapes the digital data after the ADC is ignored.

In a conventional ADC system, ADC wordlength (i.e., N) at a constant size will be delivered to the baseband for each input cycle. However, in ReADC, this wordlength size can change. In an 8-*bit* to 12-*bit* ReADC, the wordlength varies from a minimum of 8 *bits* to a maximum of 12 *bits*. The baseband must know

exactly the size of each wordlength sent by the ReADC. This is very crucial at the later stage where the digital data is converted to analog via DAC. If the baseband expects a 10-*bit* data string, but receives an 8-*bit* or 12-*bit* data string instead, there will be an error in transmission or the data will be processed incorrectly, as illustrated in Fig. 5.14.

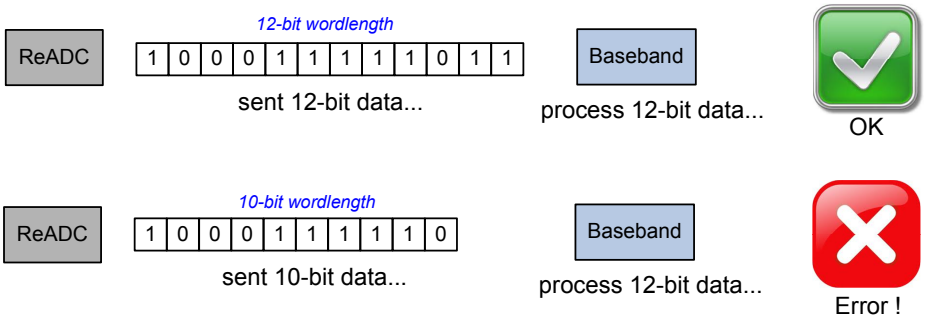


Figure 5.14: Data-passing between ADC and baseband block

So far, this issue has not been discussed in the literature analysed in Chapter 3. In the SDR implementation, the change in the wordlength can be notified to the baseband at the same time ReADC changes its N value. However, the baseband must use a different hardware configuration each time to receive a different ADC wordlength size. These changes in hardware or software configurations and settings should be considered as a part of the ReADC system. The changes carry additional reconfigurable overheads. In certain cases, if the overhead is not implemented properly, it might exceed the main ReADC module in terms of power and area penalties.

To solve the problem of data passing between ReADC and the next functional modules, the first few LSB *bits* of the ADC wordlength are truncated when not operating at the maximum resolution, N_{max}, as shown in Fig. 5.15. No additional reconfiguration schemes for the baseband are required when expecting digital data from the ADC. Furthermore, the encoding will be much simpler where the encoder for the truncated bit can be set to idle. The truncation process is a simple method used in digital design to reduce wordlength [157].

During the operation of lower resolution, i.e. $N < N_{max}$, the truncated bits will have no SA value. As a result, using this truncation approach, the computational complexity is reduced. In hardware, this will lead to a reduction in the power consumption.

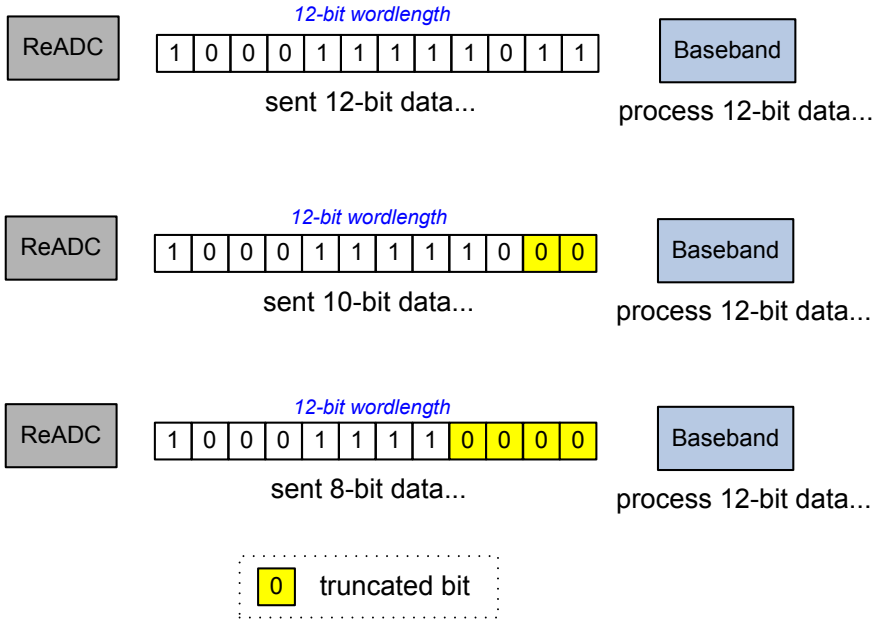


Figure 5.15: Data-passing between ADC and baseband block with the wordlength truncation method

5.7 Application of the Adaptive Algorithm in Pipeline ReADC

The pipeline architecture is chosen to implement the adaptive algorithm. This is due to its scalable configuration that enables linear power scaling. The details of the reN algorithm are also presented. The study will discuss the algorithm and the approach in detail. The system is simulated using MATLAB to demonstrate its functionality. The results are analysed and discussed.

5.7.1 System Setup

The generated UMTS signal is used as the test signal in this simulation. This signal also has FS range of $2V$ with a maximum and minimum signal fluctuated in the $\pm 1V$ range. The analog test signal is fed through the ReADC. Digital data obtained by this digitisation process are used for the SA computation to reconfigure the ReADC.

The pipeline ADC utilises a granular flash block [158–160] and is suitable for ReADC implementation. The adaptive algorithm is implemented as a control unit of a pipeline ReADC. Figure 5.16 illustrates the proposed ReADC system. It contains of N pipeline stages for an N-*bit* ReADC. Each stage is connected to the configurable switches to switch the block ON and OFF. An adaptive control unit (ACU) controls the switches for all stages. The ACU reads and processes the digital output of the ADC for each sample.

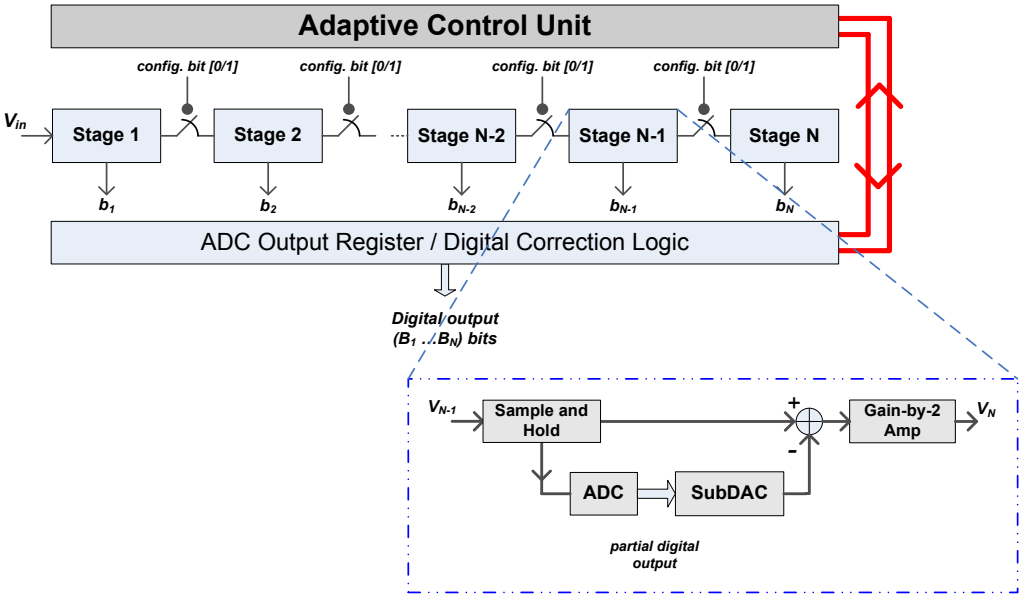
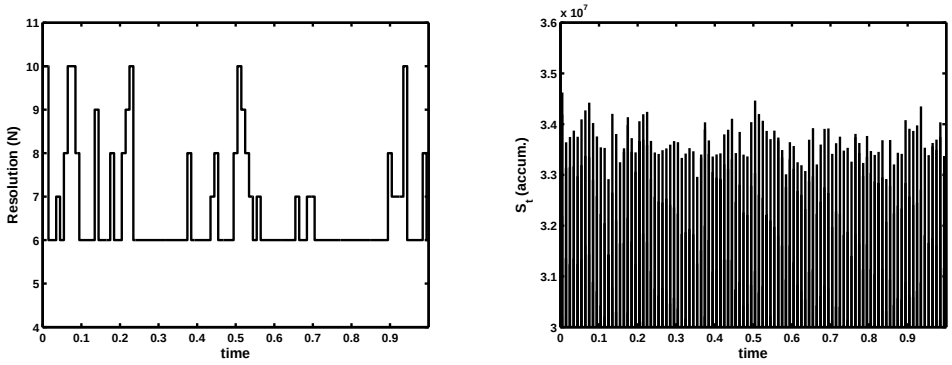


Figure 5.16: Pipeline ReADC with the ACU



(a) Variation of N for seven time slots

(b) SA result of a 6-*bit* to 10-*bit* ReADC for seven time slots

Figure 5.17: BER against N of UMTS signal for SF= 64

5.7.2 Adaptive Resolution

Algorithm 2 for a pipeline ReADC describes the adaptive mechanism for reN in detail. Figure 5.16 illustrates the architecture of an adaptive pipeline ADC for reN. The resolution of the ReADC varies from 6 *bits* to 10 *bits* depending on the fluctuation of the amplitude. If the ReADC operates at 6 *bits*, then the last four LSBs are set to zero. In contrast, when the converter operates at full resolution, then all the bits are active. Initially, ReADC begins its operation with a 10-*bit* conversion. After each $T_{measure}$, the decision of changing N is made and consequently, this decision is held until the end of the next measurement cycle.

The variation of N in the time domain as observed in seven time slots is recorded in Fig. 5.17(a). Since the threshold values are set within full resolution range, N varies from 6 *bits* to 10 *bits*. The variation is a similar pattern to the SA result presented in Fig. 5.17(b). This can be clarified further by overlapping the SA and N results, as shown in Fig. 5.18. Although there are few occasions where S_t shows slightly higher results at N_{min} than at other time during this operation, this S_t value does not exceed a vt_4 that could increase the resolution.

Algorithm 2 Adaptive Algorithm for Reconfigurable Resolution

1: Initialise parameters; set threshold values vt_1, vt_2, vt_3 and vt_4
2: Start converter with *N-bit* resolution
3: Measure the difference in SA, α of two successive digital outputs of ADC B_i and B_{i+1} within an interval $T_{measure}$ of 4 UMTS time slots ($n = 4 \times 10240$ samples)
4: Accumulate the SA of digital outputs within this interval $T_{measure}$ of n samples
5: Compare the total SA values, S_t at the end of each interval $T_{measure}$ with the threshold values, vt_l
6: **for** $N_{min} = 6 < N < N_{max} = 10$ **do**
7: **if** $S_t \geq vt_1$ **then**
8: **return** $N = 10$
9: **else if** $vt_1 > S_t \geq vt_2$ **then**
10: **return** $N = 9$
11: **else if** $vt_2 > S_t \geq vt_3$ **then**
12: **return** $N = 8$
13: **else if** $vt_3 > S_t \geq vt_4$ **then**
14: **return** $N = 7$
15: **else if** $S_t < vt_4$ **then**
16: **return** $N = 6$
17: **end if**
18: **end for**
19: ADC operates with the calculated resolution from step 6 to 18 for the remaining UMTS frame (T_{idle})
20: Continue conversion with resolution as step 3 to 19

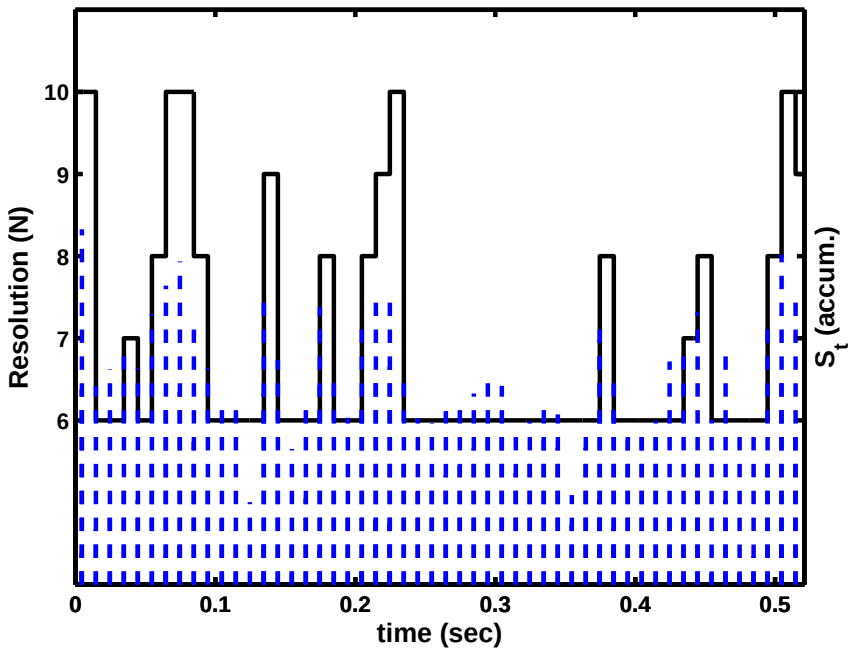


Figure 5.18: Variation of SA and N of a 6-*bit* to 10-*bit* ReADC for seven time slots

5.8 Bit-to-Error Rate Evaluation

The robustness, efficiency, and effectiveness of the adaptive algorithm are evaluated by comparing the effect of changing N with the BER of the reconstructed signal. For UMTS applications, the BER requirement is specified by the 3GPP documentations.

5.8.1 Simulation Setup

The received data in the baseband are compared with the transmitted data for the BER comparison. As illustrated in Fig. 5.19, the baseband digitises the signal using a fixed 10-*bit* DAC. At the receiver front-end, the resolution of ADC is changed from 6 *bits* to 14 *bits* and the recovered data is compared with the

original data. The range of N is extended to 14 *bits* to increase the observed window of the effect of varying N value on the BER.

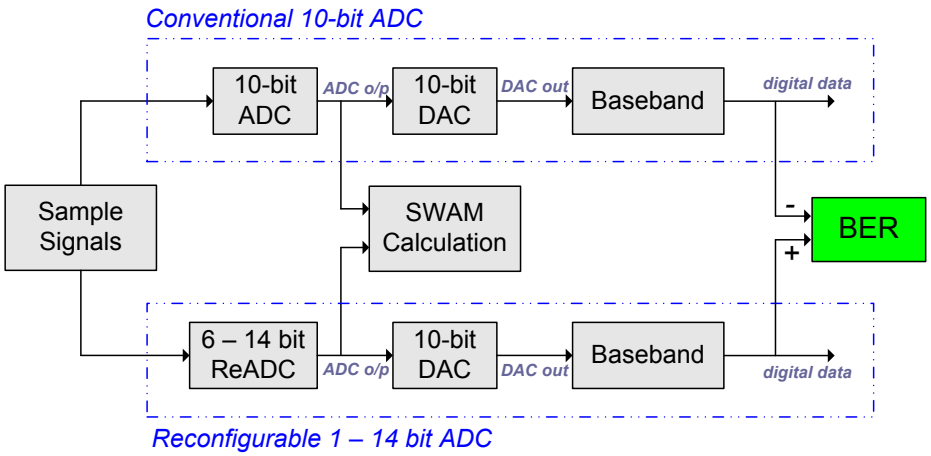


Figure 5.19: System setup for MATLAB simulation

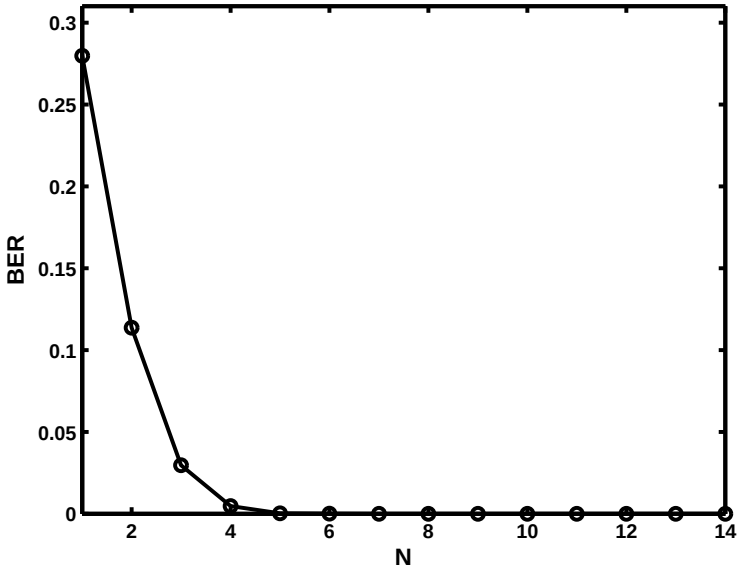
5.8.2 BER Result for Different N Values

The BER results against the variation of N are recorded in Fig. 5.20. Figure 5.20(b) is a logarithmic presentation of the BER of Fig. 5.20(a). As N varies from 14 *bits* down to 1 *bit*, the BER becomes worse. For the receiver, the requirement of BER below 10^{-3} must be fulfilled [30]. Therefore, from the graph, the value of N_{min} is 5 for SF= 64. However, $N = 6$ is chosen due to the BER being well below 10^{-3} to ensure the requirement is preserved.

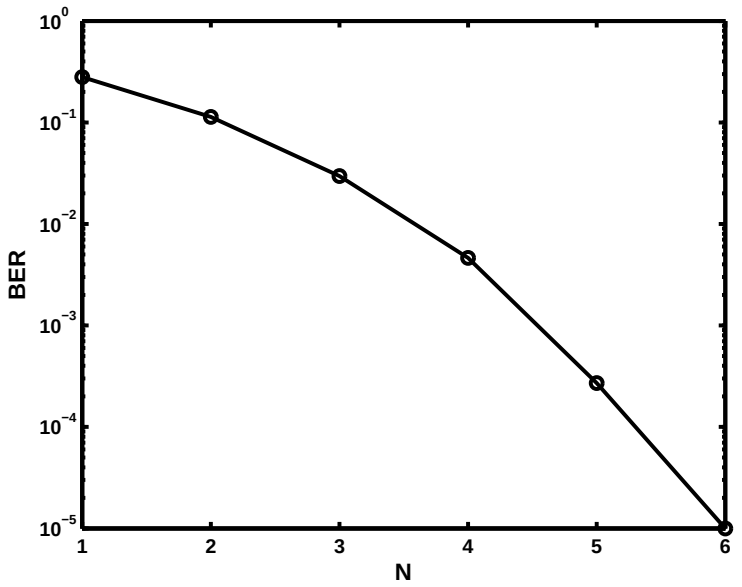
The error occurs in this simulation as a result of reducing the resolution of the ADC. In other words, the quantisation error at the lower N value is highly dominant and cannot be ignored.

5.8.3 Discussion

Since there is a correlation between SA value and the amplitude scaling, the algorithm is used to adapt at different amplitude range. Generally, when the amplitude range is reduced in a frame duration, this implies a low SA during this



(a) BER vs. N for $SF = 64$



(b) BER vs. N for $SF = 64$ (log. scale)

Figure 5.20: BER against N of UMTS signal for SF= 64

period. Hence, the converter is allowed to operate at a lower N value.

Furthermore, since the signal is evaluated for each frame, the decision to change is unique for that particular frame. This will ensure that any variation for other frames will not affect the operation of the current interval. Moreover, a constant observation period of a frame enables the signal to be evaluated periodically and effectively. This concept is similar to the analysis of the FFT, since the observed window is set to a particular period where the longer the period, the better the result could be obtained.

The implementation of the adaptive algorithm, however, must also consider four design issues. Firstly, since the SA computation involves an ADC wordlength where each bit is important, N_{xt} is set for the entire ADC wordlength. It is also possible that a fine step change in amplitude is less favourable. Therefore, computation can only consider N_{xt} for the first few MBS bits. Similarly, if a small step change in the amplitude is the main interest, N_{xt} is set to the last few MBS bits.

Secondly, in this design, $T_{measure}$ is set to half of the frame duration. It is possible that the interval of $T_{measure}$ could be set to less than that amount by further investigation. If this period can be reduced, then the response and effectiveness of the algorithm could be increased. As the idle duration of the algorithm is increased, then less time is taken for $T_{measure}$ to compute the SA result. However, the reduction of $T_{measure}$ must not alter the result. This issue will be investigated further in the next chapter.

The third case considers the threshold setting to ensure adaptivity. Since there is a correlation between SA value and the amplitude variation of the signal, the threshold values are derived from this result. However, at the moment, the amplitude reduction is chosen to be at most 30% from the maximum value or equivalent to half-power reduction. This has set the range of S_t to be within 1% of the maximum S_t computed by the SWAM module. This issue may be investigated further in the future to reach more accurate results.

The thresholds could also be set by a simple mathematical approach by considering the minimum and maximum value of S_t of a fixed 10-*bit* ADC. These values are normalised to the maximum possible SA result at a period of $T_{measure}$ of n samples. This approach, although adequate, is less accurate than the first approach, where an exact S_t result is considered. The effect of different threshold settings will be discussed further in the next chapter.

The encoding technique for the ReADC is also analysed. Other modules must know the effect of varying the ADC output. To avoid a complicated solution that could lead to hardware complexity, a truncation method is used. In this case, when the converter operates with the resolution less than maximum then the corresponding bits are truncated and assigned to zero. Therefore, other adjacent modules see this as a '0' bit.

The BER is an important indicator for observing the performance of a system. It underlines the strict specifications that need to be fulfilled to ensure correct operation. By considering the effect of varying N values on the BER performance, the minimum value for the ReADC is determined. It is expected that higher N values contribute to better BER performance. Nevertheless, the relationship between N and BER has enabled the adaptive algorithm to work within the specific requirement of UMTS.

The reduction of SA is a significant result since it affects the total power consumption P_{total} of a CMOS circuit as expressed by (5.12), which consists of three different parts: switching power $P_{switching}$, short-circuit power P_{SC}, and leakage power $P_{leakage}$, as described in [161, 162].

$$\begin{aligned} P_{total} &= P_{switching} + P_{short\ circuit\ (SC)} + P_{leakage} \\ &= (\frac{1}{2} S_w C_{load} V_{dd}^2 f) + (I_{SC} V_{dd}) + (I_{leakage} V_{dd}) \end{aligned} \tag{5.12}$$

where S_w is the SA factor, I_{SC} is the short-circuit current, and $I_{leakage}$ is the leakage current. The product of $S_w C_{load}$ is known as switched capacitance, and S_w defines the number of gate output transitions per clock cycle.

Therefore, in this pipeline ReADC implementation, the total power consumption is reduced not only by varying N values, but also as a by-product of lower SA results. However, since the analog part dominates the power consumption for a pipeline ReADC, then the reduction of power by low SA is less significant. However, for other ADC architectures and different applications, a lower SA value could be influential. Finally, with an algorithm implemented as adaptive controller, reN is able to adapt to the amplitude variations.

5.9 Summary

This study focuses on the reconfigurability issue of ReADCs. Consequently, the SWAM algorithm for a reconfigurable converter is proposed and its adaptive functionality has been verified successfully. The algorithm manipulates the ADC digital output activity and regulates the converter resolution through an adaptive digital controller unit. The increment and decrement of N is determined by threshold values vt_l and the length of the measurement period $T_{measure}$. As an implementation, the controller is designed for a pipeline ReADC. However, other typical converter types, such as flash, SAR, and $\Sigma\Delta$, may benefit from this algorithm since the main ADC unit involves only small circuit changes that include simple configuration switches. Simulation results demonstrate the algorithm is simple and the feedback loop implementation requires low computational complexity. For 6-*bit* to 10-*bit* ReADC, the average power saving of a pipeline ReADC is approximately between 10 and 40 %. In this case, the range of N_{max} and N_{min} defines the maximum reduction. The results have also verified that the algorithm is real-time reconfigurable according to signal variation and power scalable with N.

As an extension of the analysis to the proposed adaptive algorithm, the effects of parameter settings are evaluated in the next chapter.

Chapter 6

Effects of Parameter Settings on the Adaptive Algorithm

Contents

6.1 Introduction

The performance of the proposed adaptive algorithm in Chapter 5 relies effectively on its parameter settings, composed of the following:

- The measurement interval $T_{measure}$ and consequently, the number of samples n in this interval, and
- The threshold values.

Therefore, finding the right values for the parameters is important to optimise results and improve the algorithm efficiency. The optimisation of parameters has

been used widely in search algorithms, such as evolutionary computation [163,164]. The purpose of the study is to investigate the effect of each parameter on the algorithm that contributes to the final solution. In this chapter, a detailed analysis is made according to factors that determine the SA values of the ADC outputs. Consequently, proper solutions for a specific application can be found to achieve more precise and efficient results.

6.2 Parameter Tuning

The parameter space of the adaptive algorithm is evaluated in terms of (i) different UMTS time slots (i.e., the number of samples n in a period of $T_{measure}$), and (ii) the relationship of the SA and the variation of the signal amplitude that leads to setting of the threshold values. In addition, the dependent of the BER on the variation of SF at the baseband is also observed.

With different sets of parameter settings, the SA value is determined by selecting a variable to observe the impact on the algorithm and to vary its values. Generally, other parameters are fixed at constant values. For example, when determining the effect of $T_{measure}$ on the SA value, $T_{measure}$ is changed from 1 to 14 UMTS slots in a frame. Meanwhile in this analysis, the threshold values vt_l are set according to the initial observation of the conventional ADC at a specific $T_{measure}$. Similar to the previous chapter, N_{xt} is fixed as 10 $bits$ to consider the entire wordlength. The impacts of parameter variations on the adaptability of the pipeline ReADC are analysed.

6.2.1 Duration of $T_{measure}$

A UMTS frame consists of 15 time slots. The SA responses for different lengths of the slot are studied. The sample of UMTS signal is normalised to the observed time slot. For example, in the first simulation, the SA result is computed only during the first time slot of the UMTS frame ($T_{measure} = 1$ slot). No computation

is made for the remaining 14 slots of the frame ($T_{idle} = 14$ slots). The observation only considers the maximum $T_{measure}$ up to 14 slots and 1 slot for T_{idle}. Figure 6.1 illustrates the description. The SA is measured again during the first time slot of the next frame and the process remains until all the frames are used. The average value of S_t, $\langle S_t \rangle$ for the first time slot is determined, and the results are normalised to the maximum theoretical SA value, $S_t(MAX)$ for that specific time slot. If n is equal to 5120 samples for a slot, then for a 10-*bit* ADC, $S_t(MAX)$ is equal to 5119×1023, as given by (5.11).

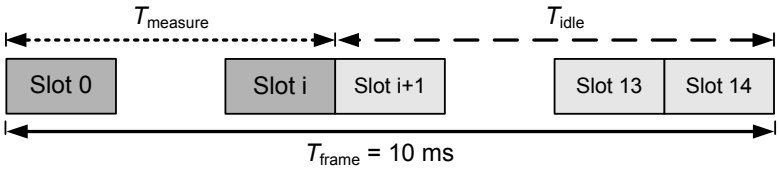


Figure 6.1: SA computation for different UMTS time slots

The parametric simulation of varying the length of $T_{measure}$ can be described in detail by considering Fig. 6.1. The UMTS frame duration is divided into two main intervals, $T_{measure}$ and T_{idle}, mentioned previously in the last chapter. During the measured interval $T_{measure}$, samples from the ADC outputs are processed to decide the change. Meanwhile, T_{idle} is the period in which the converter operates with the newly computed resolution value during the duration of $T_{measure}$.

Since each frame composes of 15 time slots, the SA result is computed by varying $T_{measure}$ for different lengths of slots. Consequently, n for the SA computation can be determined by analysing the SA values for different UMTS time slots. The first sample is taken at the beginning of the frame, and the measurement stops at the end of $T_{measure}$. For each slot, n is equivalent to the ratio of the slot duration T_{slot} and the sampling rate duration T_s. This is presented by (6.1) as

$$n = \frac{T_{slot}}{T_s} \tag{6.1}$$

where T_{slot} is $666.67\mu s$. Usually, a sampling rate four times faster than the chip-

rate is considered. Thus, f_s is $15.36MHz$, where n is equivalent to $10,240$ samples in a slot. Furthermore, since $T_{measure}$ is fixed at a specific interval, this will benefit the hardware implementation. The cyclic nature of the measurement interval makes the hardware realisation very practical, because the hardware can operate with a finite and periodic clocking sequence.

If $T_{measure}$ duration is reduced but the SA result is constant for the rest of the frame, then the adaptive response can be increased. In other words, the aim of varying $T_{measure}$ for different time slots is to identify the minimum value of $T_{measure}$ that can give a consistent SA result.

It is important to determine a specific duration for the observed time $T_{measure}$ for SA computation. Practically, $T_{measure}$ should be kept as short as possible. This is because the internal operation of an adaptive circuit can be kept to a minimum, and the converter operates with a N value determined during this period. The SA value measured during $T_{measure}$, however, it should also represent the value for the entire frame. In other words, the SA value must be constant after a certain period of time within the same UMTS frame.

6.2.2 Threshold Values and Ranges

In the previous chapter, the correlation between the SA values and the variation of signal amplitude of the UMTS was used to set the threshold values. However, the reduction in signal amplitude is considered up to 30% or an amplitude reduction to half power only. In this study, a different range is chosen and compared with the previous result. For this simulation, $T_{measure}$ is set to the value optimised from the previous section.

Previously, the thresholds could also be set using a simple mathematical approach using minimum, mean (average), and maximum values. However, this approach is not considered in this section since the result is similar to the initial approach that uses the SA correlation result. Furthermore, the correlation analysis result shows a direct relationship between the SA value and the signal

amplitude that makes the result more meaningful. In this case, the thresholds are set by directly referring to the graph (Fig. 5.11). The SA result against the variation of amplitude is recorded. Similarly, the variation of N against time slots for different ranges is also presented and discussed.

6.2.3 Influence of Different SF Values on the BER Performance

In Chapter 4, the BER is a function of G_p, which consists of spreading gain G_s and coding gain G_c. The spreading gain, which is the ratio of the chip rate R_c to the symbol rate R_s, is also known as the SF. Therefore, as SF is reduced, it is expected that the value of BER will also drop. Thus, in this section, the effect of changing SF value on the BER is observed. Since changing N at a specific SF value contributes to different BER results, thus by varying the SF value, N_{min} values are deduced at the accepted BER level. These will specify the least possible N values at different SF for the ReADC to preserve the required BER.

6.3 Results and Discussion

This section will discuss the effect of varying UMTS time slots and threshold values on the SWAM algorithm performance. The results of BER for different SF values are also analysed.

In particular, the experiments investigate the impact of changing these parameters on the adaptability of the algorithm as the values of the parameters are increased. The results for all frames are collected, and the average value is determined. The variations of N and S_t for reN are analysed.

6.3.1 Adaptive Response for Different UMTS Time Slots

We seek to find the minimum number of slots for the SA computation that represents the value for the rest of the frame. The number of time slots required for

the SA computation is determined. In this section, the response of adaptive N and S_t for different time slots are observed and discussed. For this simulation, the algorithm is set with the threshold values as in Table 6.1, and the amplitude is reduced up to half power. The sample of the UMTS signal consists of over 200 frames. At the end of $T_{measure}$ for each frame, the N value for the rest of the frame duration is determined based upon the SA computation. Total results for the entire signal are accumulated and the average value is calculated. At each simulation, the duration of $T_{measure}$ is increased from 1 slot to 14 slots. Altogether, there will be 14 simulations, and the average value is computed for each simulation. The results of average resolution $N_{average}$ and S_t are recorded.

Table 6.1: Threshold values setting for variation of time slots

threshold (vt)	**value**
vt_1	0.467
vt_2	0.466
vt_3	0.464
vt_4	0.463

6.3.1.1 Average Resolution Result for Different Time Slots

The response of the adaptive algorithm is presented in Fig. 6.2. From the figure, it is found that the average resolution of the entire sample, $N_{average}$ reaches relatively a constant value after four slots. A $N_{average}$ of 6.75 *bits* is found from the graph. There might be a slight fluctuation in the average resolution after these four slots. However, between 4 and 14 time slots, the deviation is only around $\pm 1\%$ from the average value.

6.3.1.2 SA Result for Different Time Slots

The SA result for different slots of a conventional 10-*bit* ADC is shown in Fig. 6.3. It is noted that the SA result reaches a stable value only after four time

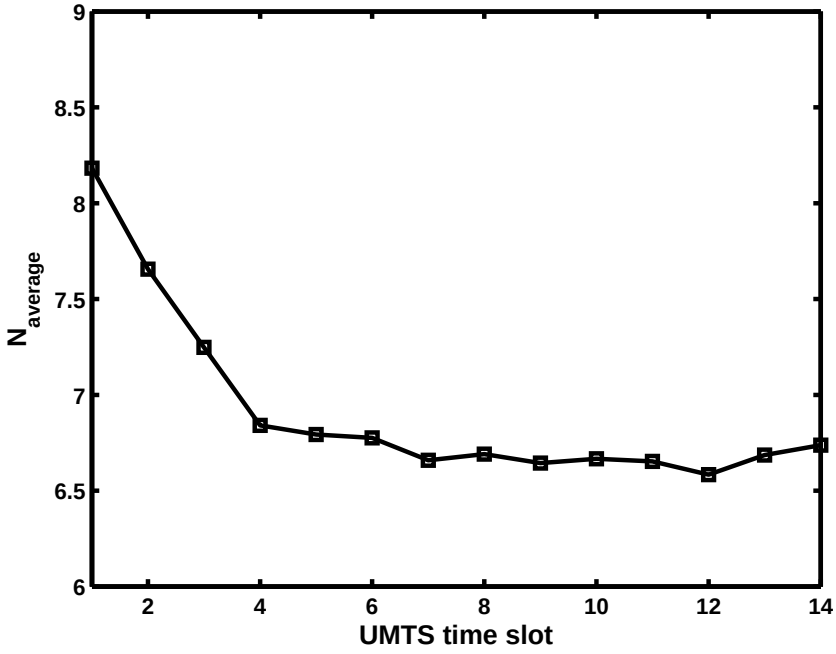


Figure 6.2: $N_{average}$ against UMTS time slots

slots. This value is approximately 46.6% of $S_t(MAX)$. Meanwhile, the SA result against time slot for the ReADC is recorded in Fig. 6.4. Similar to the 10-*bit* ADC, a stable value is found after four slots. However, this value is just slightly lower than the 10-*bit* ADC, which is about 46% of $S_t(MAX)$.

The SA comparison results for different time slots between ADC and ReADC is illustrated in Fig. 6.5. From the observation, it can be concluded that the minimum duration of $T_{measure}$ is four slots since for both cases the SA values reach constant states after four slots. This finding is sufficient to represent the SA result for the entire frame.

The SA result for the ReADC is slightly lower, since the resolution adaptively varies between 6 *bits* and 10 *bits*. When the ReADC operates at a lower N value, the truncated bits carry no SA value. Although the SA result shows only a little difference between ADC and ReADC implementations, it becomes evident when $N_{average}$ result is considered. The ADC operates at the full resolution of 10 *bits*,

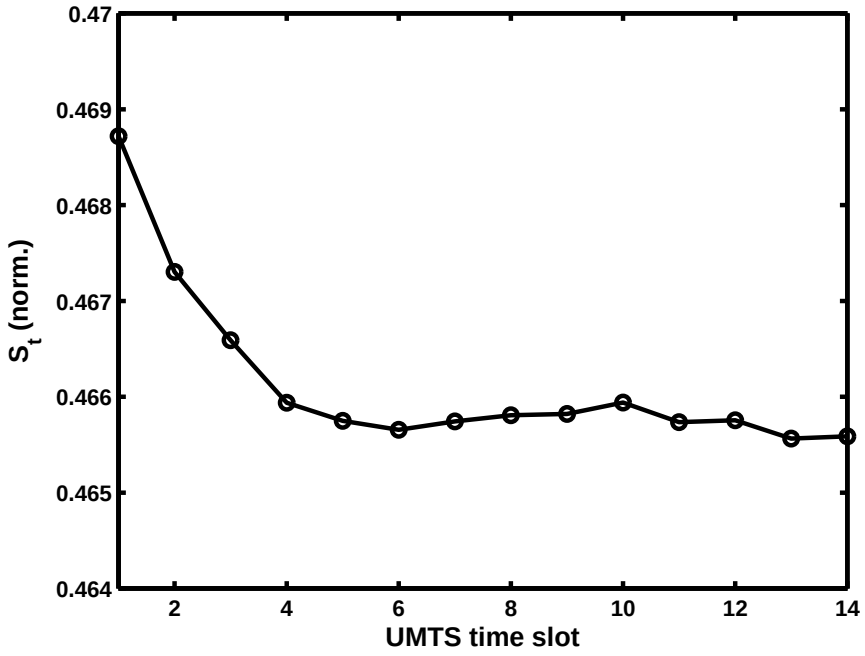


Figure 6.3: SA result against time slots for a 10-*bit* ADC

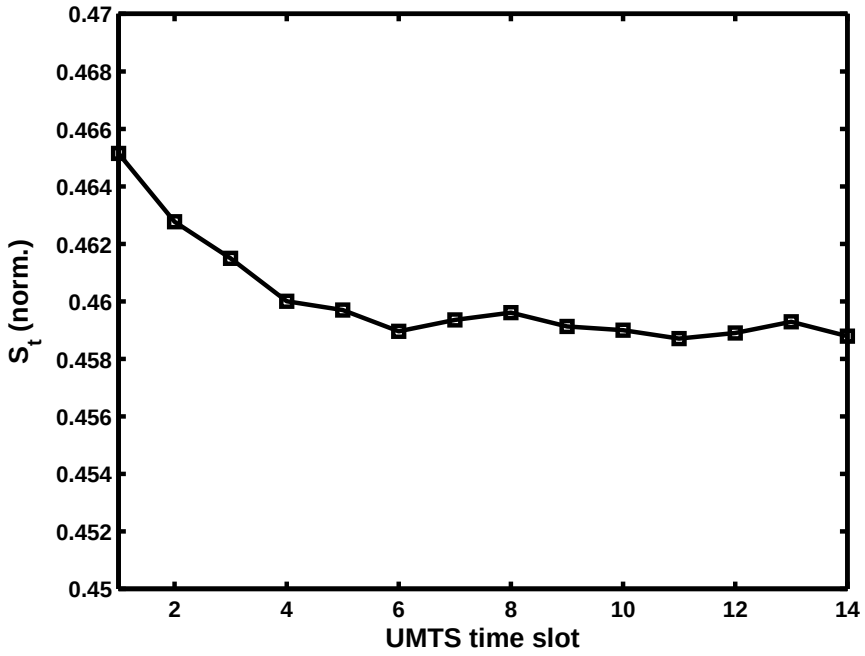


Figure 6.4: SA result against time slots for a 6-*bit* to 10-*bit* ReADC

while ReADC has an average value of 6.75 *bits*. In pipeline implementation, this implies as much as 32.5% of the average power is saved by using a ReADC. This is a favourable result for the power-constraint devices, because more power savings are translated into a longer operation time.

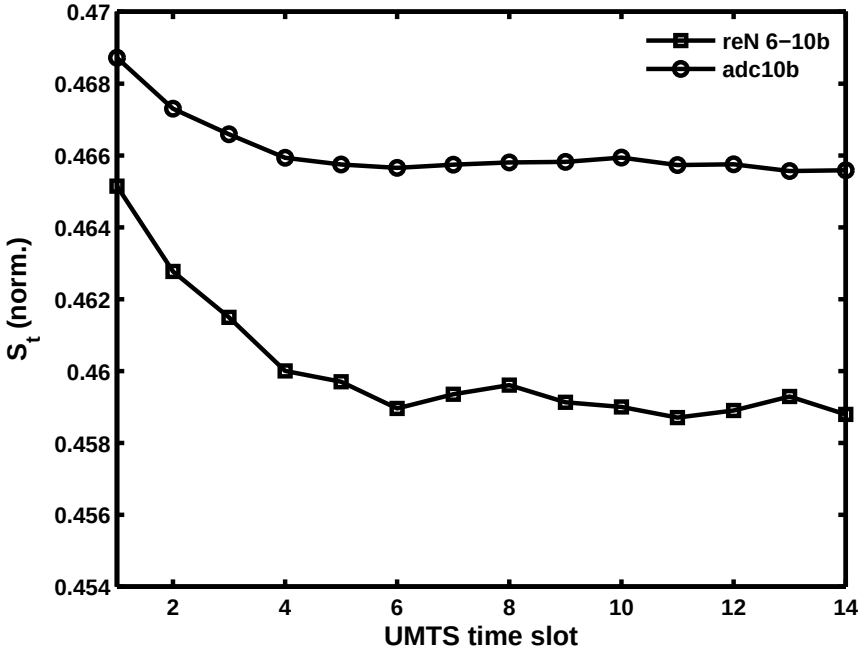


Figure 6.5: SA comparison of a fixed and reconfigurable ADCs

The normalised results, however, are not suitable for hardware implementation since fixed-point operations are used. Therefore, accumulative SA results, $S_t(accum.)$ will be used instead. The accumulative results, as shown in Fig. 6.6, indicate that the maximum value accumulative SA value, $S_t(accum.\ max.)$ is linearly proportional to the number of UMTS slots. For example, the maximum accumulative value for five time slots is nearly 2.4×10^7 changes. This is equivalent to $S_t(MAX)$ multiplied by with the maximum measured SA value, $S_t(max)$.

In conclusion, the minimum duration for $T_{measure}$ to reach a constant value that represents the S_t result for the rest of the frame is four UMTS time slots. With an adaptive implementation, the internal operation of the ADC has decreased by

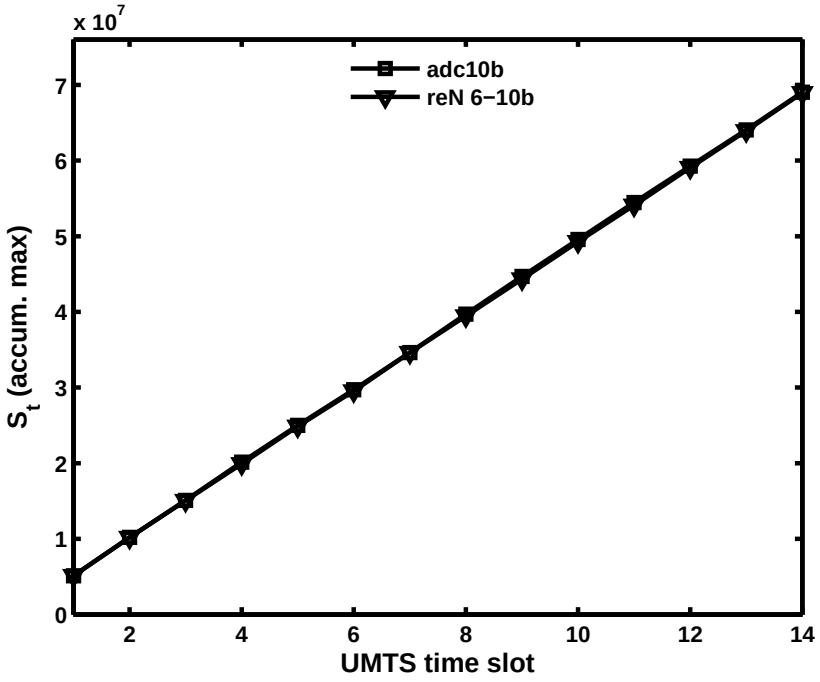


Figure 6.6: Maximum SA result for different time slots: ADC and ReADC

1.5% as observed from the SA computation. Since the results in Fig. 6.2 are also lower in $N_{average}$ compared with a conventional ADC, the reduction of ReADC power consumption is twofold: lower internal SA operation and lower $N_{average}$ value. This will translate to better power savings compared with a conventional ADC. The reduction can be increased further by setting different values of the threshold that reflect a lower $N_{average}$. However, the BER must be fulfilled. In this case, the minimum resolution of an adaptive ADC should not be smaller than 6 $bits$, considering ideal case implementation.

6.3.2 Effect of Different Threshold Values and Ranges

The relationship of the SA and variation of UMTS amplitude for four to seven time slots is recorded in Fig. 6.7. The results for different slots are consistent in shape. These graphs also confirm the conclusion from the previous section. The figure is used as a reference to monitor the effect of different threshold settings on the algorithm.

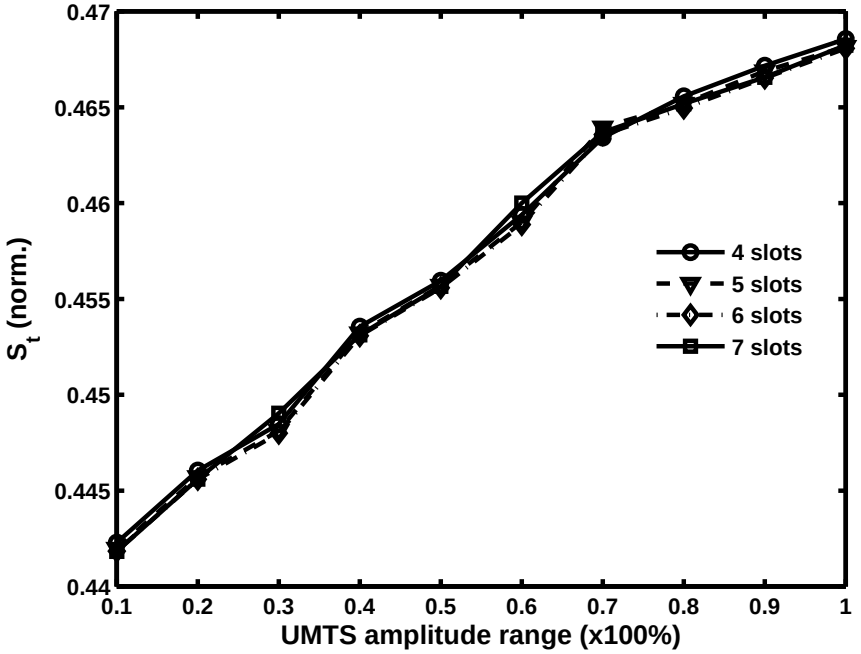


Figure 6.7: SA result against variation of amplitude for four to seven time slots

Table 6.2: Settings for different threshold values and ranges

threshold	*value*	
	range 1	range 2
vt_1	0.467	0.466
vt_2	0.466	0.464
vt_3	0.464	0.461
vt_4	0.463	0.456

Different threshold value settings of the adaptive algorithm have different effects on the responses. Table 6.2 shows two different sets of values and ranges of the threshold. The first range, called 'range 1', was implemented in the previous section. This setting covers around 1% from $S_t(max)$ or is equivalent to 19.2% of the observed S_t $(norm.)$ range. This implies up to 30% of the amplitude reduction of the signal or $-3dB$ of power (as in Fig. 6.7). As a comparison, the setting of the second range, 'range 2', covers 5% from $S_t(max)$ (the threshold values are set to be within $\pm 2.5\%$ from the middle value of this range) or equivalent to 46.15%

of the observed SA range. This setting allows as much as a 50% reduction in the analog amplitude or equivalent up to a quarter power ($-6dB$) to be monitored and adaptively changes the resolution.

6.3.2.1 Average Resolution Result for Different Time Slots

Figure 6.8 shows the results of $N_{average}$ against UMTS slots. The responses of both threshold settings are compared. For the first range, $N_{average}$ is recorded as 6.75 $bits$. Meanwhile, $N_{average}$ is 7.6 $bits$ for the second range. In Section 5.6.3, statistically, $S_t(avg)$ is 0.463 for a 10-bit ADC. Therefore, it is noted that for the second case, the adaptive mechanism mostly responded to threshold values greater than vt_3 since $N_{average}$ is greater than 7 $bits$. This threshold sets the resolution boundary between 7 $bits$ and 8 $bits$. The SA drops below the vt_4 level only on a few occasions. In contrast, lower $N_{average}$ is found for the 'range 1'. This is indicated by the adaptive response of 'range 1' since the lowest threshold value vt_4 is set to 0.463. Therefore, the adaptive mechanism responds to all possible N values for the first range.

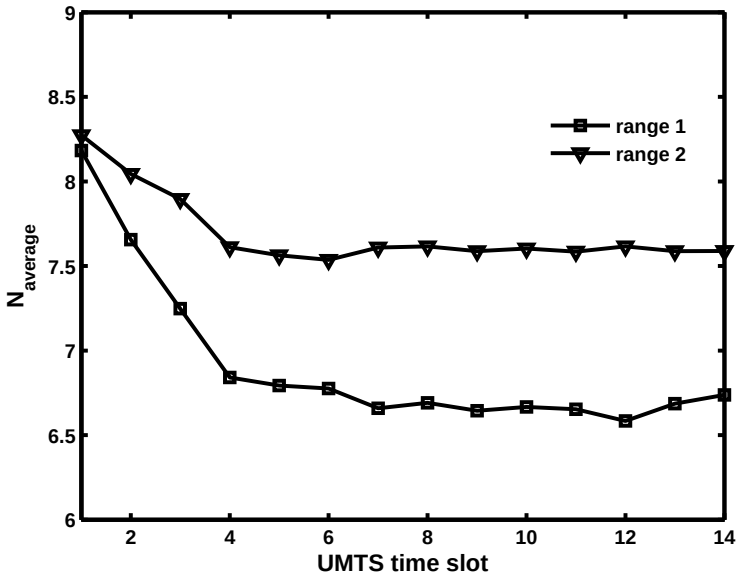


Figure 6.8: $N_{average}$ against time slots for different threshold ranges

6.3.2.2 SA Result for Different Time Slots

The effect of different threshold settings can be observed further through considering the SA response for both ranges, as depicted in Fig. 6.9. As compared to 'range 1', the SA result for the second range is higher where the average SA value reaches 0.461 after four slots. This value is approximately 0.5% higher than the first range. The result for the second range is slightly higher since $N_{average}$ is also higher than the first range.

Throughout the simulation process, the results consistently indicate that the minimum observed time slot is four to achieve a reasonable and consistent result. Therefore, it can be concluded that the minimum period of $T_{measure}$ is four slots, or equivalent to 10,240 samples per slot at a sampling rate of $15.36MHz$. Meanwhile, T_{idle} is 11 slots or as much as 73% of whole UMTS frame duration.

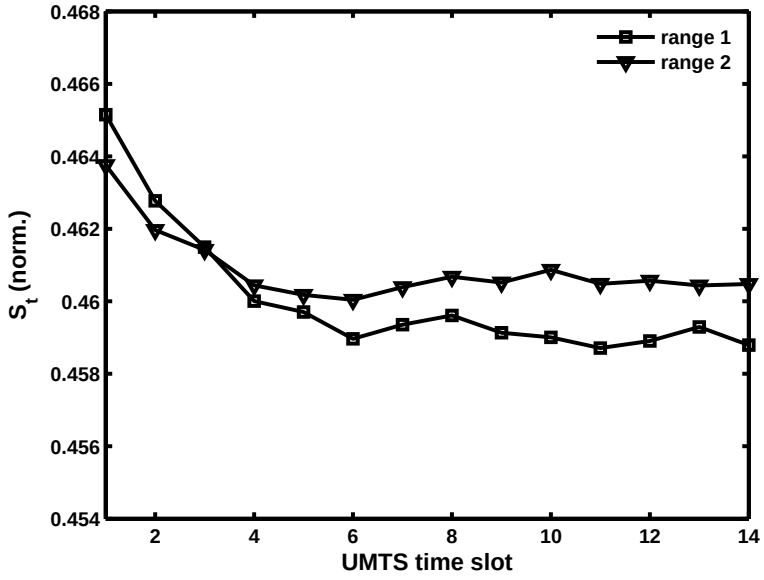


Figure 6.9: SA results against time slots for different threshold ranges

6.3.3 Effect of Different SF Values on BER Against N

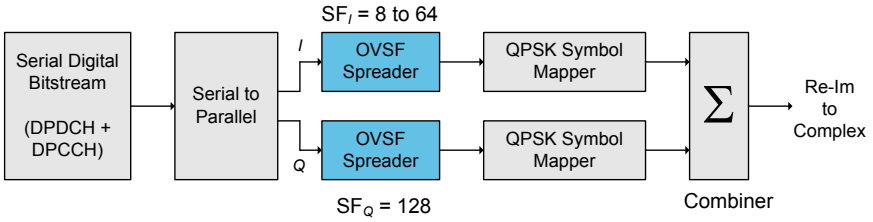


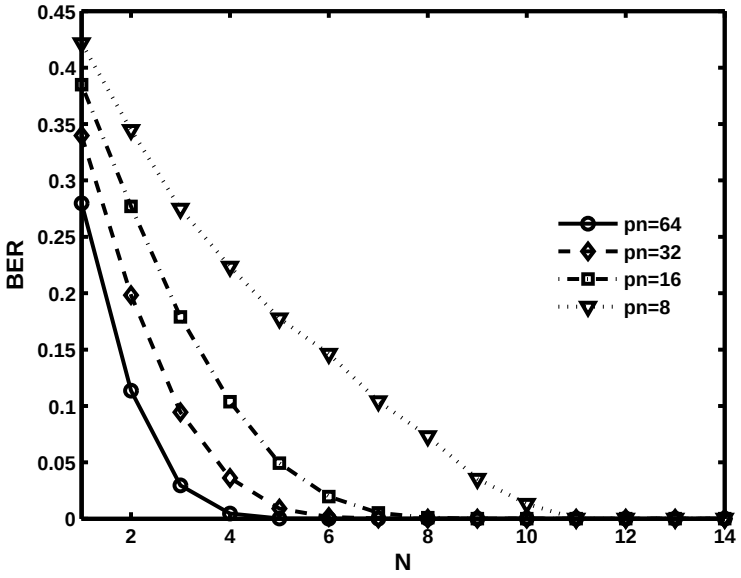
Figure 6.10: Simulation setup for BER vs. N for different SF values

Although the length of pseudonoise or PN sequence is not related to the adaptive algorithm, there will be effects of different PN length on the adaptive results. As UMTS utilises the spread spectrum technique, the symbol bit R_s is spread with the PN length. The spread data bit is also known as R_c. Thus, the PN value is equal to the SF.

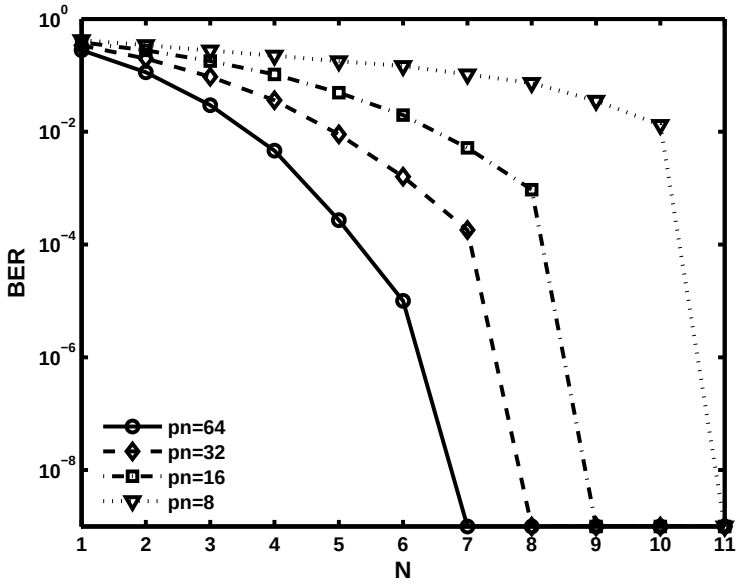
In this section, the effect of changing SF is observed. A simulation setup for changing the SF value of the I-channel at the baseband is shown in Fig. 6.10 and the BER performance is observed. The SF value of the Q-channel is retained at 128. For the BER measurement, the setup is similar to Fig. 5.19 in Chapter 5.

The graphs displayed in Fig. 6.11 are observed. As SF is decreased from 64 to 8, the BER performance also decreases, as can be seen in Fig. 6.11(a). To maintain the BER performance, the resolution needs to be increased to a higher value if SF is decreased. By considering the result in the logarithmic scale in Fig. 6.11(b), to fulfil the requirement of BER better than 10^{-3}, N values should be at least 9 *bits* and 11 *bits* for the SF values equal to 16 and 8, respectively.

In summary, the value of SF has strong effect on the minimum resolution N_{min} of ReADC. As SF is decreased, the N value must be increased to overcome the error introduced by the converter. To preserve the required BER, N_{min} for different SF values is summarised in Table 6.3.



(a) BER vs. N with variable SF value



(b) BER vs. N with variable SF value (logarithmic scale)

Figure 6.11: BER against N for different SF values

Table 6.3: N_{min} for different SF values

SF	N_{min} [**bits**]
8	11
16	9
32	8
64	6

6.4 Summary

The study of parameter space on the adaptive algorithm has given clear information on the role performed by each parameter towards the adaptability of the algorithm. For the sake of completeness, the effect of different SF values on the BER performance is also scrutinised.

The measurement duration $T_{measure}$ of the SA to determine an N value is extracted from the result of SA against UMTS time slots. In the study, it is found that the minimum duration of $T_{measure}$ is four UMTS time slots. For the sampling rate of four times the signal bandwidth, n is 40,960 samples.

Equally important are the threshold values setting controls $N_{average}$ and N_{min} throughout the conversion. The thresholds are set by observing the SA result of the conventional ADC. The settings must also take into account the BER performance to ensure the requirement is fulfilled.

These values are highly influential on the response of adaptive algorithm towards signal fluctuation. In other words, the threshold values control the adaptive variation according to signal condition. Therefore, selecting the right values for vt_l is important.

In addition, at the baseband, SF is important in data protection, as SF= 64 gives better immunity against noise than other lower values. Consequently, a high SF decreases the N_{min} value, and the requirement for higher resolution ADC can be relaxed.

Finally, from the investigation, it is concluded that to achieve the best perfor-

mance for the adaptability of the algorithm, the following parameter values are suggested:

- The minimum value of n is equivalent to four UMTS time slots ($4 \times T_{slots}$)
- To cover full-scale adaptive resolution between 6 $bits$ and 10 $bits$, vt_l is set to a range of 30% of signal reduction that includes $vt(max)$ and $vt(min)$ as Table 6.4 shows.

Table 6.4: Maximum and minimum threshold values for adaptive algorithm

threshold (vt)	**value**
$vt(max)$	0.467
$vt(min)$	0.463

- To preserve the required BER, the values of N_{min} for different SF settings are summarised in Table 6.5.

Table 6.5: Minimum N values for different SF settings

SF	N_{min} **[bits]**
8	11
16	9
32	8
64	6

In the next chapter, the SWAM algorithm is implemented in ASIC to evaluate the performances in terms of timing, area, and power. Consequently, the overhead introduced by the ACU is analysed. The hardware module synthesised by the algorithm is discussed, and its impact on the ReADC system is investigated.

Chapter 7

Control Unit Implementation of Adaptive Algorithm

Contents

7.1 Introduction

This chapter presents the implementation of an adaptive algorithm for the ReADC developed in Chapter 5. The behaviour of the algorithm is coded and the functionality is verified using Verilog HDL. The ACU module for the reN is developed. The module is synthesised using UMC 0.18μm CMOS technology. The timing and area results are evaluated. Similarly, the performance in terms of power consumption is examined.

We begin the chapter with a suggested power estimation model for reN used to measure the reconfiguration overheads. For the sake of completeness, the estimation model for reFs is also included. The models evaluate the overhead introduced by reN and reFs separately, using the information of ReADC design at the system level.

7.2 Power Estimation for ReADCs

Normally, a conventional ADC consists of two main blocks. The first block is an analog unit (AU), such as flash, pipeline, SAR, $\Sigma\Delta$, and so on. Meanwhile, the second block is a digital unit (DU), performing encoding, error-correction, register, decimation process, or as digital output buffer. Figure 7.1 shows operational block diagrams of ADC and ReADC.

(a) Block diagram of conventional ADC (b) Block diagram of ReADC

Figure 7.1: Conventional ADC and ReADC operational blocks

Considering Fig. 7.1(a), generally the power consumption of a conventional ADC can be written as (7.1).

$$P_{ADC} = P_{Analog\,Unit\,(AU)} + P_{Digital\,Unit\,(DU)} \quad W \tag{7.1}$$

In the reconfigurable approach, since an ACU is required, as illustrated in Fig. 7.1(b), the power consumption of ReADC can be written by adding the term P_{ACU} as shown below:

$$P_{ReADC} = P_{Prog.\,Analog\,Unit} + P_{Prog.\,Digital\,Unit} + P_{ACU} \quad W \tag{7.2}$$

The ACU is used to trigger the configuration process. In general, it sends the configuration switches in the same concept as FPGA.

7.2.1 Power Estimation of Reconfigurable Resolution

ReADC is a modified ADC module, as illustrated in Fig. 7.1(b), and consists of a programmable analog unit (PAU) and a programmable digital unit (PDU). An ACU completes the feedback loop that performs measurement, evaluation, and decision for the ReADC architecture. Therefore, (7.2) can also be written as

$$P_{ReADC} = P_{PAU} + P_{PDU} + P_{ACU} \quad W \tag{7.3}$$

Since the PAU contains extra but simple configurable switches compared with an analog unit of a conventional ADC, then the P_{PAU} can be approximated to P_{AU}, the power of the analog unit. Likewise, P_{PDU} is relatively equal to P_{DU}, the power of the digital unit, since a simple truncation method is used. Furthermore, a digital unit is included for encoding, error correction, digital register or output buffer for an ADC and consumes relatively less power than an analog unit does. Similarly, since the PDU only includes a small extension part for reconfigurability, then the P_{PAU} is much greater than the P_{PDU}. Moreover, the ACU performs more complex digital functions than the PDU, thus in this case, P_{ACU} is relatively much larger than P_{PDU}. By considering these conditions, (7.3) is compared with (7.1) and the relative power approximation is

$$\frac{P_{ReADC}}{P_{ADC}} = 1 + \frac{P_{ACU}}{P_{ADC}} \quad W \tag{7.4}$$

A pipeline ADC is used in wireless communication due to its medium to high N, and it operates with medium to fast f_s [22, 120, 165]. It consists of a number of stages that are directly proportional to N. Each stage has a minimum of one comparator. With a common implementation of 1.5 *bits* per stage, two comparators are required for each stage. Therefore, in a pipeline ReADC case with a maximum

of *N-bit* resolution and with active operational stages m, the power dissipation is

$$P_{ReADC} \approx N.\left(\frac{m}{M}\right).P_{1-stage\,pipeline} + P_{ACU} \tag{7.5}$$

$$P_{ReADC} \approx N.\frac{m.(\kappa.P_{comparator})}{M} + P_{ACU} \quad W \tag{7.6}$$

where M is the total number of stages, which is effectively 1 *bit* per stage, $P_{comparator}$ is the comparator power, and κ is power coefficient according to other circuit elements, such as SH, sub-digital-to-analog converter (sub-DAC), and so on. Since the comparators dominate the power of each pipeline stage, then the power for each stage is approximately equal to $\kappa.P_{comparator}$. Thus, for a pipeline ReADC, the resolution is reduced by switching off stages completely, starting with the LSB. From (7.4) and (7.5), two cases can be considered. Firstly, the ratio predicts how much the power is contributed by the ACU relative to a conventional ADC. Secondly, by keeping the ACU at minimum, the power consumption can be varied linearly with N.

Meanwhile, since DSP is used, the power consumption by a granular pipeline ReADC estimated in [120] is

$$P_{ReADC} = \left(\frac{N}{M}\right).P_{block} + P_{DSP} + P_{SH} + P_{DEC} \quad W \tag{7.7}$$

where P_{block} is the power of single pipeline stage, P_{DSP} is the power consumption of DSP, P_{SH} is the power consumption of sample-and-hold circuit, and P_{DEC} is the power of decoder unit.

7.2.2 Power Estimation of Reconfigurable Sampling-Rate

As the ReADC requires an ACU to perform the adaptive operation correctly, the power consumption is given by a generic equation:

$$P_{ReADC} = P_{PAU} + P_{PDU} + P_{ACU} \quad W$$

At the system level, by considering V_{dd} and the total current I_{total} that flows into a circuit, the power consumption is given by

$$P = V_{dd}.I_{total} \quad W \tag{7.8}$$

Any application using an analog circuit has to make sure that the capacitances (input, output, or intrinsic) have to be fully charged and fully discharged for the system to work properly. A charge of Q_c coulombs of the largest capacitance C must be stored within the sampling interval t_s, which is $1/f_s$ given by

$$Q = I.t_s \quad C \tag{7.9}$$

and

$$Q = C.V_{dd} \quad C \tag{7.10}$$

By combining (7.9) and (7.10) and substituting in (7.8), the power consumption is given by

$$P = \frac{C.V_{dd}^2}{t_s} \quad W \tag{7.11}$$

Since C and V_{dd} values are fixed, P will increase when $t_s$decreases. This result implies that for higher f_s, the power increases. To gain better insight, each element in (7.3) is substituted with (7.11) and rewritten as

$$P_{ReADC} = \frac{C.V_{dd}^2}{\hat{t}_s}.(a_1 + a_2 + a_3) \quad W \tag{7.12}$$

where a_1, a_2, and a_3 are power factor coefficients for the PAU, PDU, and ACU, respectively. Meanwhile, the power consumption for an ADC can be written as

$$P_{ADC} = \frac{C.V_{dd}^2}{t_{sample}}.(b_1 + b_2) \quad W \tag{7.13}$$

where b_1 and b_2 are power factor coefficients for analog and digital units, respec-

tively. Similarly, $P_{PAU} \approx P_{AU}$, $P_{PDU} \approx P_{DU}$, and $P_{ACU} > P_{PDU}$. Therefore, by considering $a_1 \approx b_1$ and $a_2 \approx b_2$, (7.12) is compared with (7.13) to give

$$\frac{P_{ReADC}}{P_{ADC}} = \frac{t_{sample}}{\hat{t}_{sample}}.\left(1 + \frac{a_3}{a_1 + a_2}\right) \quad W \tag{7.14}$$

Two cases can be considered from (7.14). Firstly, the ratio reveals how much the power is dissipated by the ACU in reFs mode relative to a conventional ADC. Secondly, by keeping P_{ACU} at minimum, the equation reveals how much the power can be saved when reducing the f_s value. In this case, maximum P_{ReADC} occurs when $\hat{t}_{sample} = t_{sample}$.

7.3 Design and Implementation of Adaptive Control Unit

The realisation of an adaptive algorithm for reN into ASIC involves three design stages. Firstly, the algorithm is transformed into hardware architecture. The algorithm is analysed and appropriate circuit topologies are considered. Then, the architecture is simulated at the register transfer level (RTL) to verify the functionality. Finally, the design is synthesised and later, it is placed and routed. Consequently, the performances are evaluated in terms of speed, area, and power consumption.

7.3.1 Methodology and Design Flow

The ASIC design in this chapter has adopted the design flow described in Section 2.8. In this chapter, the design involves architectural simulation (i.e., RTL), gate level synthesis, and physical implementation.

Using MATLAB simulation as elucidated in Chapter 5, the algorithm is implemented in floating-point arithmetic. Hardware design, however, requires fixed-point implementation since hardware resources are limited. Therefore, the opera-

tion and reference values are implemented in an integer operation. The hardware design is coded using Verilog for both RTL simulation and synthesis. A CMOS technology library of $0.18\mu m$ is used for both gate level synthesis and physical implementations.

To test the functionality of the hardware, the output of a 6-*bit* to 10-*bit* ReADC from the system level simulation is converted to a text file. The file that contains the binary representation of the UMTS signal is used as the input of the ACU, and the performance is evaluated.

7.3.2 Architecture and Operation

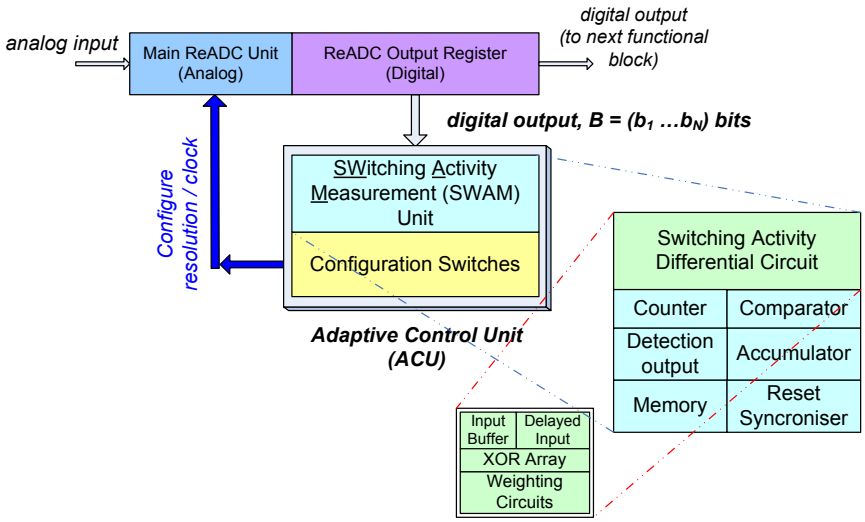


Figure 7.2: ReADC architecture

The binary-weighted ACU architecture is separated into two parts, as shown in Fig. 7.2. One part is used to measure the SA of the ADC output. This module is known as the SWAM unit. Another module is called the configuration unit (CU), and its operation is controlled by the configuration signal from the SWAM unit. This signal carries information for the switching configuration of reN operation.

The SWAM module involves a detection process, and it is the main part of the architecture. It consists of the SA measurement circuit, an accumulator, a

digital counter, a digital comparator, and memory cells. A reset synchroniser is also included to reset the circuit during power-on state.

The SA measurement circuit is developed using XOR gates and a simple weighting circuit. The accumulator uses the counter to collect a specific ReADC output of n samples in the duration of $T_{measure}$. Meanwhile, memory cells are built using D flip flops (DFFs) and used to store an intermediate computation result and the threshold values. The digital comparator evaluates the difference between SA and the thresholds.

The outcome of the comparison process determines the configuration switches for reN. In a pipeline ReADC implementation, the reconfigurable switches will activate any pipeline block that corresponds to the configured N.

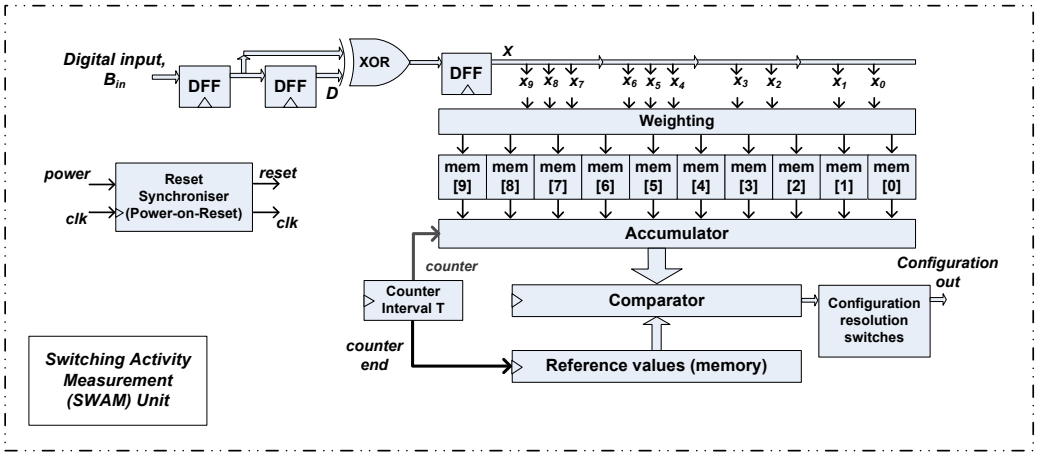


Figure 7.3: Hardware architecture of binary-weighted ACU implementation

To compare the same behaviour of the UMTS signal between system level simulation and the hardware implementation, the digital outputs of the ADC from the MATLAB simulation are recorded in a text file. The text file contains a binary representation of the UMTS signal for each sample. Three digit hexadecimal (hex) numbers are used for each sample since a maximum of 10 $bits$ is used for the digitisation process. This file is used as an input for the hardware architecture to observe its functionality. Each sample will be read by the hardware at every

clock cycle until the end of the sample.

Figure 7.3 shows detailed hardware circuitry of the ACU. At each clock interval, the SA of the ReADC output, B_{in} (parallel data) is measured by considering the binary-weighted hamming distance between the current converter wordlength and the previous wordlength. Since all *bits* of the ReADC are considered, 10 parallel XOR gates are used for the parallel ReADC output. The weighted hamming distance result for each gate x_i is connected to a weighting circuit, which simply consists of memory cells connected directly to each XOR output. The number of 'ones' from XOR gates is computed, and an intermediate result is stored in the memory cells, $mem[0]$ to $mem[9]$. This memory is updated at each clock cycle, and this intermediate result is collected by the accumulator for a duration of $40,960$ samples ($n = 40,960$). This is equivalent to $4 \times T_{slot}$ length. Altogether, as much as $10,240$ samples for each T_{slot} are processed. At the same time, the counter measures the number of samples, and the 'end-of-counter' signal is triggered when the required samples are collected.

Initially, the memory cells are loaded with reference values, thresholds vt_1, vt_2, vt_3, and the lowest threshold vt_4, as described in Table 5.5. Since the $S_t(MAX)$ is $27F5C01_{hex}$, all threshold values are set by multiplying $S_t(MAX)$ with the normalised values.

At the end of each interval $T_{measure}$ in a frame, the result of S_t is compared with the threshold values. A configuration signal is triggered from this process. If S_t is higher than vt_1, then the signal is sent to the CU to operate with a full resolution. Likewise, when S_t is lower than vt_4, the CU will be loaded with a control signal to operate with a minimum resolution. Similarly, the converter will vary the value of N adaptively when S_t value is in-between these minimum and maximum threshold values.

7.4 Configuration Scheme for Reconfigurable Resolution

In this section, the configuration mechanism for reN is explained. The CU for the reN consists of reconfiguration switches. Figure 7.4 illustrates how the reN is implemented. The example considers a reN adaptation of 6-*bit* to 10-*bit*. A configuration switch uses 16-*bit* data preloaded with the sequence '0000_0011_1111_1111' or $03FF_{hex}$ for 6-*bit* operation. The first two MSBs of the sequence are neglected, and the next ten MSBs are used to control the configuration switches of the pipeline ReADC. Meanwhile, the last four LSBs are used as an intermediate location for the shifted data. This data are shifted to the left when N is increased and shifted to the right when N is reduced, as shown in Fig. 7.4. Therefore, for 6-*bit* conversion, the control sequence is also represented by $03F_{hex}$, which the final F_{hex} is ignored.

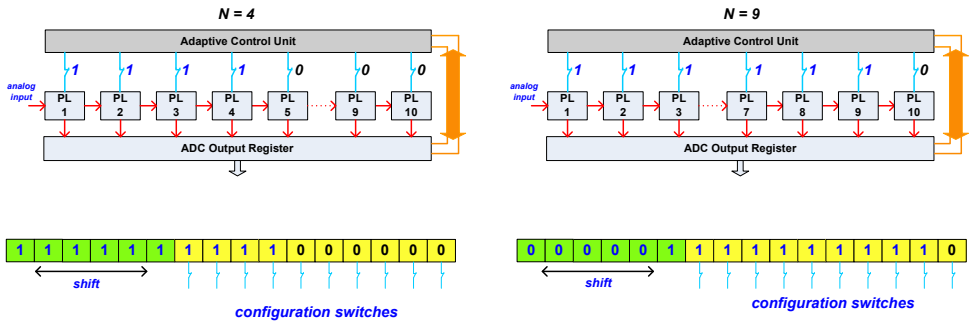


Figure 7.4: Configuration scheme for reN

7.5 Results and Discussion

7.5.1 RTL Simulation

Figure 7.5 shows the result of Verilog RTL simulation. Initially, the ACU triggers the configuration switches to operate with N_{min} value of 6 *bits* as indicated by

Figure 7.5: RTL simulation result of ACU for the reN

'*readc*'. This 12-*bit* switch is equivalent to '03F' in hexadecimal. Only the last 10 LSBs are used as configuration switches. In Fig. 7.5, 'b' represents the input of the ACU that reads the 10-*bit* of ReADC wordlength. Originally, a 10-*bit* ADC output is supplied to the ACU module and indicated by 'a'. Meanwhile, 'd' is the delayed input, 'x' is the output of XOR logics, and 'mem' is the accumulation process. For each ADC conversion cycle ('$endConv$'), the SA is calculated and is accumulated for a $4 \times T_{slot}$ duration. At the end of the measured duration ('$cnrT0\_end$'), the ACU compares the result with threshold values. This process occurs when the flag for the end of the measurement period ('$cnrT\_end$') is raised. The SWAM module is reset after each $T_{measure}$ duration and is idle during the rest of the UMTS frame duration, T_{idle}. The ACU shows its adaptivity by changing from 10 $bits$ ($3FF_{hex}$) to 9 $bits$ ($1FF_{hex}$), 8 $bits$ ($0FF_{hex}$), and 6 $bits$ ($03F_{hex}$). In this case, the first four LSBs of the preloaded sequence are neglected.

7.5.2 Synthesis and Performance Evaluation

The design is synthesised using a Synopsys Design Compiler with UMC $0.18\mu m/1.8V$ CMOS typical library models. In Table 7.1, the ACU for a reN ADC with a maximum operating frequency of 100 MHz consumes a dynamic power (P_{ACU}) of $1907\mu W$. From this amount, the internal power of 1461 cells consumes 97%. In terms of area, its gate density is 3090 2-input NAND equivalent $gates/mm^2$. The ACU unit (A_{ACU}) occupies 99% of the area and only 1% is used for the reset synchroniser. As much as 85% of 1461 cells is the combinational instances. The ACU consumes most of the cell internal power: 98% of $1846\mu W$. The reset synchroniser utilises the remaining internal power.

As compared with the adaptive control unit implementation in [123, 166] that consumes a total power of $1.752W$, the proposed ASIC implementation of an ACU consumes just $1900\mu W$ of power at the most. The gate density is also small, only 3090. Furthermore, the feedback loop is short since the ACU is connected directly

Table 7.1: ASIC performance of binary-weighted ACU

	ACU (Binary Weighted) using SWAM Method
Maximum frequency	100 MHz
Process technology (Supply voltage)	0.18 μm CMOS ($1.8V$)
Area (cell)	37699 ($\approx 0.038\ mm^2$)
Gate density NAND equivalent	3090
Total cell	1461
Power (μW)	Cell internal: 1846 (97%) Net switching: 61 (3%) Total dynamic: 1907 Cell leakage: 2.9

to the digital output buffer of the main ReADC module.

Table 7.2: Reconfigurable overheads of ACU compared with the existing pipeline ReADCs

ReADC	f_s **(MSps)**	A (mm^2)	P (mW)	% ($\frac{A_{ACU}}{A_{ReADC}}$)	% ($\frac{P_{ACU}}{P_{ReADC}}$)
[121]	15.36	0.35	15	10.9	12.7
[167]	50	1.2	35	3.2	5.4
[168]	$10-40$	1.15	72.8	3.3	2.6
[169]	40	1.9	72	2	2.6
[115]	80	1.9	94	2	2

Moreover, using the suggested power estimation model for reN, the power headroom introduced by the ACU module is less than 13% compared with the developed pipeline ReADCs (P_{ReADC}) with the same technology, as shown in Table 7.2. The table indicates that the area overhead due to the reconfiguration mechanism is also small, less than 11% compared with the area of the main ReADC modules (A_{ReADC}) developed by using similar CMOS process technology.

The hardware implementation of ACU for reN reveals that the architecture

utilises simple circuitry and requires only small amount of logic gates. Consequently, RTL simulation results have validated that the proposed adaptive algorithm is suitable for ASIC implementation.

Although the hardware coding using Verilog has not been optimised to achieve better area and power consumption, the results have already indicated that the overhead introduced by the ACU is very small. Considering real-time adaptability introduced by the ACU with signal variation (since it can respond to different amplitude fluctuations), the module is considerably smaller and attractive for low-power and battery-operated applications.

7.6 Summary

This chapter focuses on ASIC design, simulation, and implementation of an ACU module for a pipeline ReADC. In the beginning, power estimation models for ReADCs at the system level are suggested. The power models are presented to estimate the additional power penalty due to the reconfigurability scheme and to predict the power consumption of ReADCs, consisting of a variation of N and f_s. Then, an adaptive control module for reN is developed. The controller measures the SA the ReADC digital output, manipulates and triggers the reconfiguration switches for reN. The ASIC performance results exhibit small area complexity with gate density, around 3090 equivalent 2-input NAND gates in $0.18\mu m/1.8V$ CMOS. The power consumption of the ACU is only $1907\mu W$. These values are very small compared with the previous implementation of adaptive controllers implemented in DSP. Simulation results also verified that the ACU is real-time adaptive. The area overhead of the ACU is small, 11% at most. Meanwhile, the power overhead introduced by the ACU module is less than 13%. Both area and power overheads are measured in terms of the ratio between the ACU module and the main ReADC module.

This method for measuring the SA of the ADC wordlength is an attractive

concept because the variation of a signal can be distinguished using a simple computational approach. It is shown that reconfigurability is achieved by the hardware implementation of this method. To explore other potential uses of the SWAM method, a noise detection system concept is presented in the next chapter.

Chapter 8

Noise Correlation Analysis and Detection Circuit using SWAM Method

Contents

8.1 Introduction

Noise often limits the performance of the transmitted signals and degrades signals quality. Moreover, the stochastic nature of noise makes it difficult to predict, and hence, hard to detect. In hardware implementation, reduction of noise can only be optimised in the baseband, where complex and intensive computation is executed using the DSP. Although analog pre-filtering is applied at the receiver front-end

to reduce interferences, fractions of noise still exist due to the non-ideality of the device.

This chapter analyses a correlation between SA and noise level using the proposed SWAM method. As a result, a method to detect noise strength in a noisy signal is proposed. The functionality of the concept is verified using behavioural simulations. Consequently, a noise detection module is implemented in an ASIC hardware. This detection module provides an attractive solution for signal recovery and data enhancement.

8.2 Noise

8.2.1 Overview

Unwanted disturbances, such as environmental noise, device noise, harmonics, coupling, and crosstalk, often cause deterioration in signal quality. These are the results of disturbance natures in channel or the non-ideality of a device, leading to undesired time-varying voltage and current, and superimpose on an originally transmitted signal. In nature, noise is normally random and difficult to predict.

In wireless communication, the presence of noise is difficult to detect at the receiver front-end due to its passive analog components or limited function of analog circuits. Usually, analog pre-filtering is used to suppress unwanted frequency bands and out-of-interest signal components. Typically, a fixed time-constant is used. Although a programmable filter [170] can also be used to select a signal at a different frequency band, its configuration signal is sent from a digital block either after signal evaluation or by user-defined selection. Therefore, noise prediction and counter-measures are normally executed in basebands using DSPs, due to their complex algorithms and intensive arithmetic operations.

8.2.2 Source and Mechanism of Noise

Wireless channels are susceptible to noise and disturbances, since many forms of wireless signals share the same open-air medium. To make the typical scenario worse, noise is also present inside the receiver itself. In principle, there are two types of noise: real noise and other unwanted components. Random charge carrier movements generate real noise. Meanwhile, unwanted components arise as an effect of nonlinearity of components and circuits. Generally, both effects are referred to as noise.

To minimise chip area, weight, and power consumption of electronic devices, mixed-signal systems and single chip solution are implemented. Both analog and digital functions are merged as system-on-a-chip (SoC) or system-in-a-package (SiP). The analog part is very sensitive to noise, affecting the system performance. It must be protected from unwanted disturbance, especially from the adjacent noisy digital circuit. The digital block has better noise immunity, while the analog block is very sensitive and more prone to noise [171]. To reduce noise, several approaches have been implemented. At the semiconductor level, double and triple wells for the CMOS are used [172]. Guard ring and buried layers methods can also reduce noise coupling through substrate or electromagnetic coupling through bond wires [173, 174]. Similarly, at the circuit level, analog and digital blocks are arranged at distance, in a balanced configuration, in a symmetric topology as well as with proper grounding [175]. Likewise, filtering, encoding, and channel equalisers [176] are available methods to suppress noise at the system level.

8.2.3 Noise Measures

Due to its stochastic nature [177], noise is difficult to predict and thus hard to detect. Up-to-date estimation and detection of noise involve a probabilistic function [178] that requires complex computation, hence intensive hardware operation. Only DSP is suitable to perform such an operation, whereas implementation in ASIC seems inappropriate due to the algorithm complexity. Although noise pre-

diction and counter-measure are implemented successfully in DSPs, additional time and power is also used. Therefore, preliminary noise detection that uses available hardware resources with simple operation and at low-power consumption is highly desirable. Such a module can relax the need for complex operation in the baseband. As soon as noise is detected, the system has more freedom and greater opportunities to improve signal quality.

8.2.4 Effect of Noise on ADC output

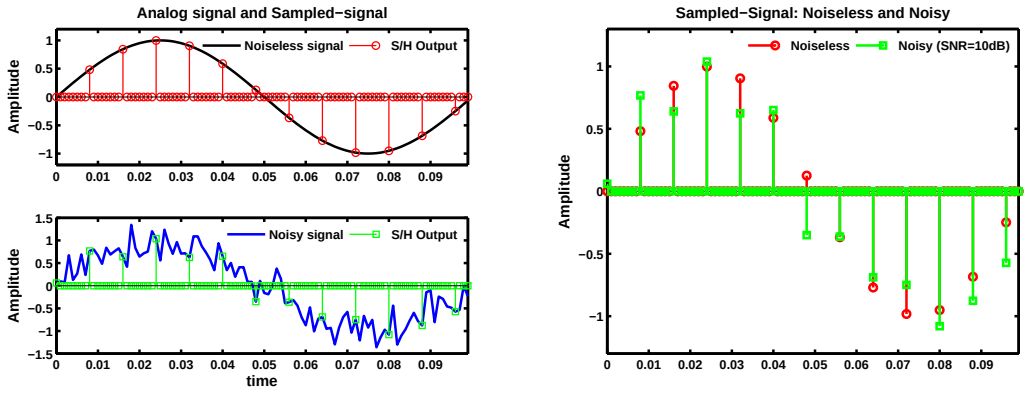
Since ADC is a transformation device, digital signals carry the same information as analog signals, except in a different domain. Therefore, any rapid fluctuation in analog amplitude is represented by a large variation in ADC wordlength. In the presence of noise with large amplitude, the effect of noise becomes eminent.

Figure 8.1 illustrates a simple example of the effect of noise on the signal, in which the amplitude of noisy signal deviates from its original shape depending on the noise strength. Although certain types of signal are in noise-like forms, the noise affects the actual amplitude and consequently, reduces its actual signal strength.

As depicted in Fig. 8.1(a) (upper figure), a noiseless signal has shown smooth continuity and the SA between samples is small. In the presence of noise, as shown in Fig. 8.1(a) (lower figure), the signal is largely fluctuating, leading to significant bit transition between consecutive samples. This effect can be demonstrated further when the SH outputs between noiseless and noisy signal are superimposed, as illustrated in Fig. 8.1(b). Magnitude differences exist between these signals and consequently have different ADC outputs. Table 8.1 presents the digital outputs of the first six samples of both noiseless and noisy signals.

The digital outputs between noiseless and noisy signals show the deviations, especially for the last four LSB bits. The MSB is equivalent to the largest step change in analog amplitude, and the LSB is equivalent to the smallest change. Thus, from Table 8.1, noise has caused the amplitude variation and contributed to

the fluctuations in ADC outputs, especially for the LSBs. Therefore, by monitoring the digital changes between sampled signals, the difference between noiseless and noisy signals can be distinguished.



(a) Noiseless and noisy signal

(b) Sample-and-hold output

Figure 8.1: Noiseless and noisy signals: analog and SH output comparison

Table 8.1: 8-*bit* ADC outputs for noiseless and noisy signals

sample	ADC output (8*bits*)	
	noiseless	*noisy*
0	B_0: 10000000	B_0: 10001101
1	B_0: 11000000	B_0: 11000111
2	B_0: 11001100	B_0: 11100010
3	B_0: 11111111	B_0: 11111111
4	B_0: 11001100	B_0: 11001000

8.3 SWAM Method for Noise Correlation Analysis

From previous observation and discussion, a noise correlation analysis that uses the SWAM method of ADC digital output is proposed for signal enhancement and preliminary noise detection. The algorithm of the SWAM method is discussed in

detail in Chapter 5. Similar to the previous implementation for a ReADC, a noise detection circuit requires the entire ADC wordlength for SA computation. Although only the first few LSBs are necessary to determine small signal fluctuation, the rest of the wordlength (MSBs) is used to measure large signal changes. In short, the ADC wordlength can be seen as being similar to the analog signal that carries two signal components, DC and AC, with DC components by MSBs and AC components by LSBs.

8.3.1 System Setup

Figure 8.2 illustrates a proposed noise detection system. The digital output of the ADC is fed into the SWAM module to calculate the SA value within an interval $T_{measure}$ on n samples. Then, SA is compared with a reference value of total SA (SA_{ref}) measured without the presence of noise (noiseless signal).

This implementation uses a conventional 10-*bit* ADC. Consequently, the entire wordlength of 10 *bits* is used for the SA calculation. Additive white Gaussian noise (AWGN) is used as the noise source in this setup. The noise is added to the UMTS signal, and the noise amplitude is adjusted to obtain different SNR values. The impact of different noise levels on SA variations is observed.

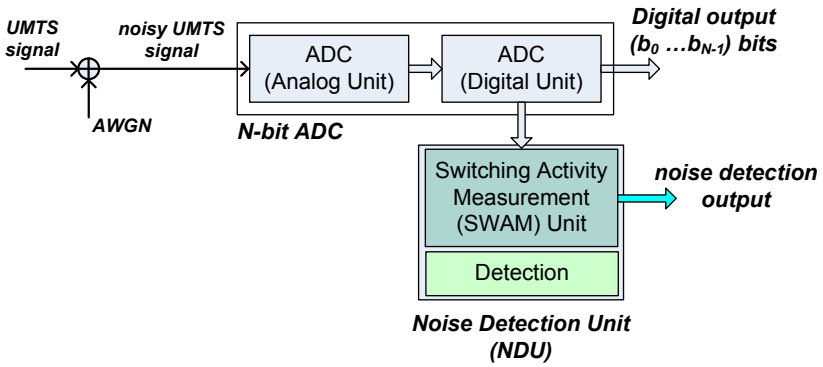


Figure 8.2: System setup for noise correlation analysis and detection

8.3.2 System Implementation

For system simulation, AWGN noise is added to the UMTS signal. The amplitude of the noise is adjusted to obtain a SNR range between $10dB$ and $30dB$. Therefore, the effect of the noise that makes the amplitude vary from 30% to 3% of the FS amplitude range is monitored.

A 10-*bit* pipeline ADC is used, and the output of the ADC is connected to a noise detection unit (NDU) module (Fig. 8.2). The SA_{ref} of the noiseless signal is measured as a reference. Then, the SA at different SNR values is measured with an increment step of $2dB$. To imitate the actual hardware implementation, the sample is accumulated for five time slots or one-third of the entire UMTS frame duration (i.e., $n = 51,200$ for $f_s = 4f_{sig}$) and the average SA recorded.

8.3.3 Simulation Results and Discussion

MATLAB simulation results of SA for different SNR values are presented. The effect of different weighted coefficients is presented for comparison. Three different types of weighting coefficients are considered: binary-weighted of 2^{i-1} (i.e., 1,2,4,8..), linear-weighted of i (i.e., 1,2,3,4..), and odd number-weighted of $2i-1$ (i.e., 1,3,5,7..) at the i-th position of each bit, for $i \in \mathbb{N}$. Consequently, the implementation results of linear-, odd number- and binary-weighting are recorded and discussed.

8.3.3.1 SA Against SNR for Different Weighting Effects

The system simulation result of the proposed noise detection circuit is presented in Fig. 8.3. The SNR is varied from $10dB$ to $30dB$ with a $2dB$ step size. The simulation is repeated 30 times and the mean values obtained. The accumulated SA value at each SNR level is normalised with the SA_{ref} of the noiseless signal. In the graph, due to high signal fluctuations in the presence of noise, small SNR value results in a high SA. The activity decreases when the SNR is increased.

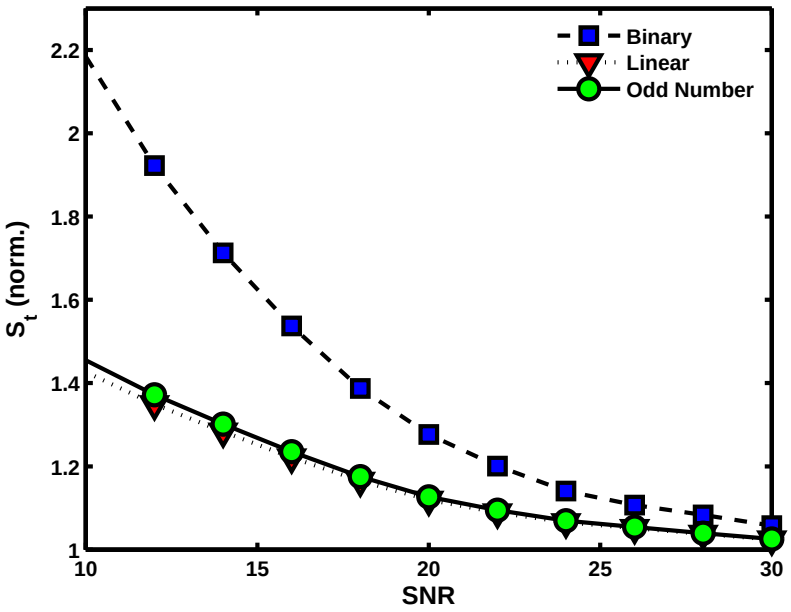


Figure 8.3: SA result against SNR for different weighting effects

Linear- and odd number-weighted algorithms give the same results as shown by the similar curves. Binary-weighted algorithms, on the other hand, show a greater degree of freedom (DoF) where the range of the result has increased by approximately 155% comparing with linear- and odd number-weighted implementations. This significant increment leads to greater DoF in noise detection. As a result, the variation of signal due to noise level can be detected with higher accuracy. Furthermore, with a larger SNR range, the advantage of weighted implementation will be significant.

The result indicates that there is a correlation between the SA result and the variation of the signal due to different noise levels. Therefore, the proposed approach could be used to detect the presence of noise in the signal. As a result, the proposed solution provides preliminary noise detection that can assist in further signal improvement.

8.3.3.2 Variance of SA Against SNR

The variance result of 30 repeated simulations of SA computation in the previous section is collected, and its mean value result is presented in Fig. 8.4. From three different weighted algorithms simulated earlier, binary-weighted implementation shows the most consistent result, as the variance is linearly proportional to the noise level. As the noise level decreases, the variation of S_t becomes smaller. Furthermore, it also gives the smallest value of the variance compared with other weighting types. This indicates that the SA computation result for a binary-weighted simulation is persistent and more reliable than the other weighting coefficients.

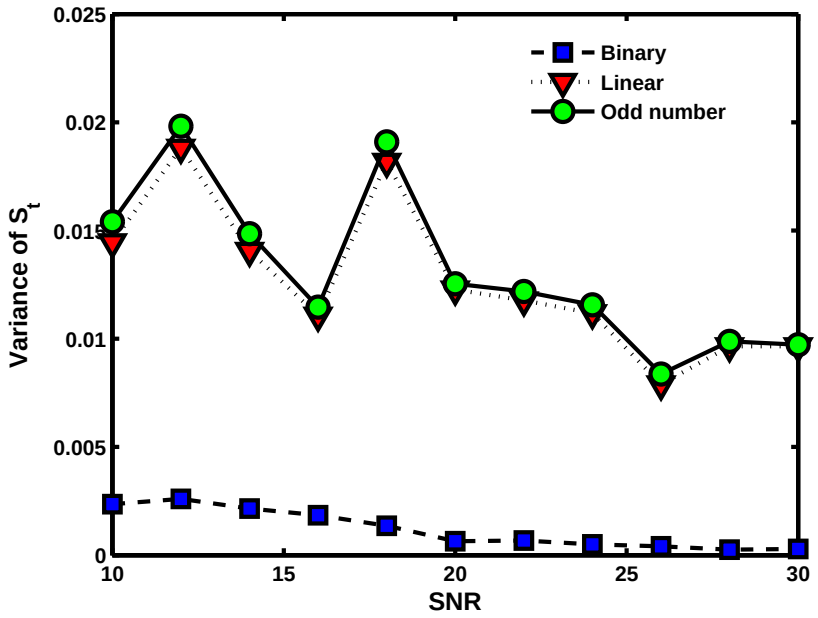


Figure 8.4: Variance of SA for different SNR values

8.3.3.3 Effect of Different f_s Values on SA Computation

The effect of varying the sampling frequency f_s is observed in this section by changing from f_s to $4f_s$. The value of f_s is $7.68MHz$, which is twice the signal bandwidth. Therefore, in this analysis, f_s is changed from $7.68MHz$ to $30.72MHz$.

Figure 8.5 shows the result for different f_s value. For all f_s values, the number of samples measured is the same. In other words, $T_{measure}$ is constant and equal to five UMTS slots for different f_s. As shown in the figure, the range of the normalised SA is one-third of the UMTS frame and is increased with the increment value of f_s. The SA value doubles when f_s is multiplied by factor of two. The result indicates that the range of SA results and consequently, the DoF can be improved with higher f_s. However, the computational complexity is also increased by a factor of two.

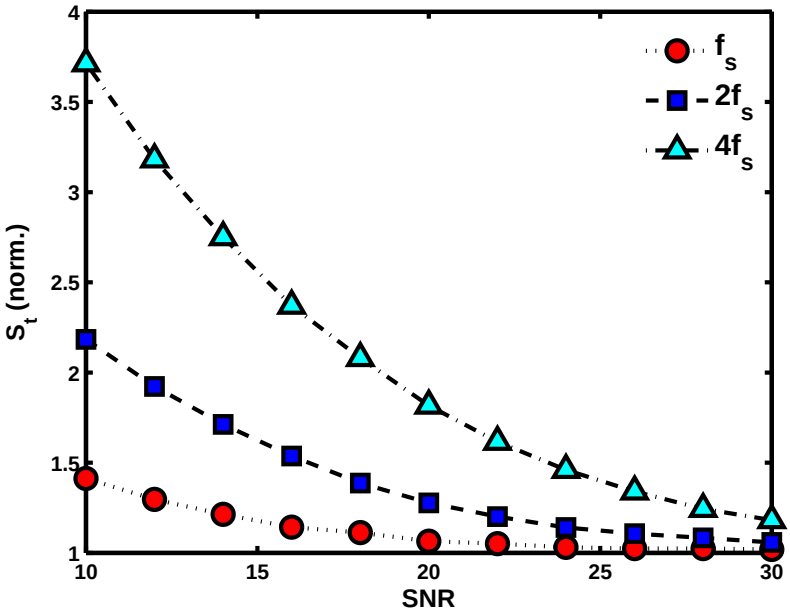


Figure 8.5: Effect of different f_s values on the binary-weighted SA computation

8.4 Hardware Implementation

To verify the concept of noise detection circuit, the ADC output of the proposed system in Fig. 8.2 simulated using MATLAB is saved into a text file. Both noiseless and noisy conditions are considered. The text files for both cases contain hexadecimal numbers that are used as the input for the hardware implementation. Therefore, the input file is the digital format of the UMTS signal from the

digitisation process using ADC. Then, the SWAM module, implemented in ASIC, reads the hexadecimal input that represents the binary format of the UMTS signal and processes the file. At first, the hardware functionality is checked with Verilog-RTL simulation. After the results are verified successfully, the module is then synthesised, placed, and routed to evaluate its area and power performances.

8.4.1 Architecture

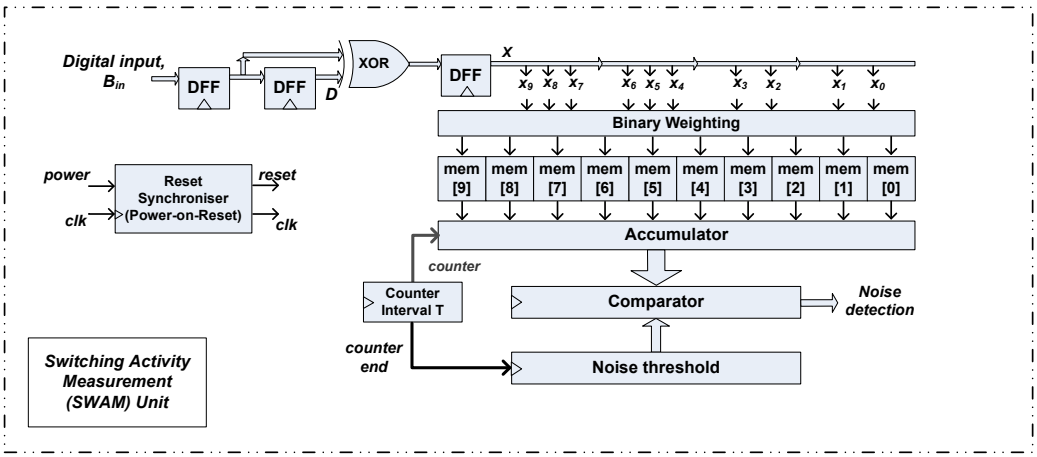


Figure 8.6: Hardware architecture of binary-weighted NDU

Figure 8.6 shows the circuit architecture of a noise module for ASIC implementation, and the module is described in Verilog. This is known as the NDU module. The input signal of the 10-*bit* data is generated using MATLAB. The current 10-*bit* input is compared with previously delayed input by XOR gates. The output of the XOR gates is accumulated using separate memory cells for each bit and simultaneously to perform as binary-weighted elements. The SA between two consecutive inputs is accumulated for the duration of five time slots. Each time the ADC converts a sampled analog signal into digital output, the counter is triggered and continues to accumulate the changes until it is reset for every interval (one-third of the frame). At the end of an interval ($cntrT\_end$), the accumulator output is compared with a reference value SA_{ref} stored in the memory (flops) at

reset. This reference value is obtained from the SA results of a noiseless signal of the same duration. The outcome of the comparison will trigger the detection flag (*noise_detect*), where one '1' is assigned if SA is larger than SA_{ref} and '0' otherwise. The process is repeated for every third of the frame duration.

8.4.2 ASIC Performance Results and Discussion

8.4.2.1 RTL Simulation

The architecture of SWAM module in Fig. 8.6 is implemented to form the NDU. Figure 8.7 shows the simulation result of the NDU architecture. Similar to the reN case, 'b' presents the input of the NDU, and 'x' indicates the output of XOR between b and the delayed input 'd'. The counter uses an end of conversion flag, '*endConv*' from the main ADC unit to accumulate the samples. When 51,200 samples are reached, '*cntrT_end*' raises the flag to HIGH ('1'). The average SA_{ref} result of a frame for a noiseless UMTS signal is 1956000_{hex}. The UMTS signal is superimposed with an AWGN of $20dB$ strength. From the previous result (Fig. 8.3), S_t is 1.3 times higher than the reference value at $20dB$. For this noise strength, the reference is set as $20F0000_{hex}$. In this case, if S_t value is lower than the noiseless reference, no noise is detected. Likewise, if the S_t is higher than the reference at $20dB$, noise is detected, and the NDU will trigger a flag to indicate that a noise level higher than SNR of this value is detected. The simulation result of the NDU shows that there is noise higher than $20dB$ after the tenth slot of the first frame and after the fifth slot of the second frame.

8.4.2.2 Synthesis Results

The architecture shown in Fig. 8.6 is synthesised, placed, and routed using the Design Compiler and Silicon Ensemble with a UMC $0.18\mu m/1.8V$ technology library. Table 8.2 presents the ASIC performance results of the NDU module.

In terms of area, NDU (A_{NDU}) consumes only $0.015mm^2$, which is of 1230

Figure 8.7: RTL simulation result of binary-weighted NDU

Table 8.2: ASIC performance of binary-weighted NDU

	NDU (Binary-weighted) using SWAM Method
Maximum frequency	100 MHz
Process technology (Supply voltage)	0.18 μm CMOS ($1.8V$)
Area (cell)	15002 ($\approx 0.015\ mm^2$)
Gate density NAND equivalent	1230
Total cell	468
Power (μW)	Cell internal: 1043 (95.5%) Net switching: 49 (4.5%) Total dynamic: 1092 Cell leakage: 1.4

2-input NAND equivalent gate density. The power consumption of the NDU (P_{NDU}) is 1092μW with 96 % of the total amount dissipated from 468 internal cells. Another 4 % of power is dissipated from a total of 475 switching nets. The maximum operating frequency is 100 MHz. The SWAM unit largely consumes the internal power of 1043μW, which is as much as 96.7%. Meanwhile, only 2.4% is dissipated by the power-on-reset (POR). From the 462 internal cells, 374 units or 81% is combinatorial logic and the remaining 19% is non-combinatorial logic.

The results show that the implementation of NDU module occupies a very small area and dissipates only a small amount of power. The module could be integrated with an ADC unit that already exists in an electronic device to preliminarily detect the presence of noise. Since the module consumes small power and area overheads, it is suitable for low-power applications in a ADC-based system. With extensive research in wireless communication towards SDR, a NDU module offers an attractive solution for signal enhancement, as well as providing reconfigurability options.

The use of weighted coefficients has increased the range for the SA margin

Table 8.3: Estimation of power and area overheads of NDU against the existing ADCs

ADC	f_s (MSps)	A_{ADC} (mm^2)	P_{ADC} (mW)	*binary-weighted*	
				$\frac{A_{NDU}}{A_{ADC}}$ (%)	$\frac{P_{NDU}}{P_{ADC}}$ (%)
[179]	100	0.8	24.2	1.8	4.5
[180]	100	0.8	25.2	1.8	4.3
[181]	100	2.5	67	0.6	1.6
[182]	100	0.98	90	1.5	1.2
[183]	100	2.5	180	0.6	0.6

greatly, and improved detection significantly. Also, the power consumption of the binary-weighted NDU implementation is significantly lower compared with the main ADC module with the same process technology that operates between $25mW$ to $180mW$, as Table 8.3 shows. In this case, the area overhead introduced by the detection mechanism for binary-weighted implementation is 1.8% at most, compared with the area of the developed ADC modules (A_{ADC}). Similarly, the maximum power overhead is just around 4.5%, by the same comparison. By introducinga NDU to the main ADC, the system is able to detect noise with only a small power overhead.

As compared with the previous chapter, the power required by the NDU is less than the ACU since only one reference value is stored inside the memory cells. To improve the range of detection, more reference values can be used that correspond to different noise levels. This requires a look-up-table (LUT) for various noise strengths. However, the power dissipation of the NDU will also increase.

8.5 Summary

The concept, simulation, and implementation of a noise detection circuit for ADC-based electronic devices using the SWAM method are presented. The SA of the ADC digital output is used to differentiate between noisy and noiseless signals. In the presence of noise, the SA of a noisy signal is larger than that of a noiseless

signal, as verified by MATLAB simulation results. The results show that there is a correlation between the SA value and the different noise strength. In this case, the SA of digital outputs varies with different SNR values, which is a high SNR value leads to a low SA value and vice versa.

In addition, the inclusion of weighting to the algorithm has also significantly increased the DoF for decision-making. At the system level simulation, a binary-weighted algorithm has increased the range of total switching activity value by 155% compared with linear- and odd number-weighted implementations. The wider margin provided by the binary-weighted implementation has improved the determination of the noise level threshold.

The ASIC performance results show that the NDU module occupies a small footprint and consumes up to a mere $1.1mW$ of power in $0.18\mu m/1.8V$ CMOS technology. With the advantages of a small area and power penalties, the NDU is attractive for low-power ADC-based applications and noise-prone device. By combining the NDU and the ADC, the system could enhance its noise detection mechanism and lead to signal improvement.

Furthermore, the Verilog code for the ASIC implementation of the module has not been optimised and can be further improved, which may result in better performance. In addition, the proposed SWAM method could also be used to configure other operational circuits to improve signal quality.

Final remarks and the conclusions of the work are presented in the next chapter. This will summarise the proposed SWAM method on adaptive control implementation for a ReADC and for noise detection systems.

Chapter 9

Conclusions and Future Work

Contents

9.1 Introduction

ReHW has been identified as an important research target in electronics design due to the advantages of development cost reduction, optimum hardware reuse, dynamic resource utilisation, and efficient computational complexity. In addition, ReHW enables real-time performance improvement and scalable power management, which are vital for eco-friendly systems and power-aware green device technology.

There is a need to use a truly adaptive ADC to respond to signal changes and reduce its power consumption with less implementation complexity. Therefore, the focus of this thesis is to investigate the reconfigurability aspects of real-time

ReADC suitable for wireless communications. Particular emphasis has been given to an adaptive control mechanism of the ReADC that changes the converter operation according to UMTS signal variations. This has paved the way for establishing requirements for future ReADC technologies.

This chapter presents a summary of the work carried out in each chapter of the thesis. The main objective of each chapter is evaluated to assess its accomplishment. The main achievements are highlighted. The whole thesis is concluded and finally, future research developments are suggested.

9.2 Summary of Thesis

Advancements in the electronic industry have compelled active researches to widen the design horizons of low-power design and mixed-signal circuit methodology. ReHW technology has become important to tackle the issues of power consumption and reconfigurability. The power issue is very significant for battery-operated applications and portable wireless devices. Besides the reduction in power consumption, the functionality enhancement of electronic devices is also seen as an important aspect. To move towards a green environment, it is crucial to reduce the power consumption of electronic devices. It is also equally important to continuously enhance the functionalities and improve the performances of the devices, while keeping power consumption at a minimum. Power performance can be improved by using ReHW technology due to the flexibility to scale its power consumption adaptively according to different operating conditions.

Up to date, existing works on ReADCs focus entirely on analog architectures and circuit design techniques, whereas the reconfiguration mechanisms totally rely upon external control signals generated by the DSP in the baseband. From a circuit architecture point of view, the developed ReADCs are similar to the conventional ADCs, except additional simple configurable switches are required by the ReADCs. Thus, the power consumption of the analog ReADC module at

a maximum resolution is approximately equal to the conventional ADC using the same process technology and power-aware circuit approaches.

Previously developed ReADCs could be improved with the integration of a digital control module that is combined together with the analog ReADC module to adaptively control the converter reconfiguration at a minimum additional power. Generally, the aims of the thesis are to investigate reconfigurable and adaptive design issues for a ReADC system and to establish a low-power solution for adaptive ADCs. Specifically, this research is an attempt to develop a low-power, single-chip, real-time ReADC system that elicits an adaptive response to signal variations and scales its power utilisation at a low reconfigurable overhead. In particular, the thesis focuses on the analysis, design, and implementation of a low-power digital controller unit for ReADCs necessary for the adaptive implementation.

In a reconfigurable system, the issues of overhead and configuration methods are normally addressed. Meanwhile, for real-time reconfigurable systems, the mechanism of 'detect-and-change' is a critical factor for determining the adaptive response of the system to a dynamically changing environment. Detailed analysis on the reconfigurability enables further developments in ADC-based applications.

The effect of time-varying signal behaviour on the adaptive mechanism of ReADCs is analysed. The behaviour of analog signals is determined from the ReADC output, where it represents an equivalent digital wordlength of a sampled analog data. An adaptive mechanism is achieved by manipulating the output of the converter. The digital outputs are processed and compared with reference values to trigger the reconfiguration switches.

Chapter 2 presents the fundamental elements of ReHW in modern electronic design. The concept and advantages that make ReHW a current research interest are discussed. Typical architectures of ReHW and existing reconfiguration methodologies are also presented. In addition, the chapter also discusses the use of ADC in wireless communication. An overview of UMTS is described along

with the typical receiver type used in UMTS/FDD mode. A Zero-IF receiver is selected normally due to its simple circuit configuration, as that is especially suitable for battery operated portable devices. Furthermore, the minimum and maximum limits of the ADC parameters are included. The minimum values for N and f_s are bounded to specific applications or requirements. Meanwhile, the maximum N and f_s values are limited by the power consumption requirement of an ADC, where the power grows with the increment of both N and f_s.

Chapter 3 reviews the existing ReADC architectures, including the state of the art of ReADCs for portable devices and wireless applications. Previous research works on adaptive control mechanism are also discussed. Existing techniques of the adaptive control of ReADCs use DSPs at the baseband to monitor and program the hardware.

Chapter 4 presents the ADC specifications for UMTS applications as a case study. UMTS documentations are used to derive the main ADC parameters in FDD mode. Zero-IF wireless receiver architecture is selected, and the derivation is developed according to this architecture. This determines the required N_{min} and $f_s(min)$ values for such an application. The research demonstrates that the minimum resolution of an ADC for UMTS/FDD mode is 10 *bits* for a practical realisation that includes design imperfections. Meanwhile, the minimum required sampling frequency is 10 Msps. These specifications have considered Zero-IF receiver architecture with $I - Q$ implementation.

Chapter 5 presents an adaptive algorithm for a pipeline ReADC using a binary-weighted hamming distance method. The proposed technique, the SWAM method, exploits the SA measurement of the ReADC digital outputs at specific observed intervals and programs the ReADC according to the reference values.

Initially, the UMTS is generated and analysed in terms of periodicity, stationarity, and framing structure. Then, the concept of the binary-weighted hamming distance of the ADC output is presented. In this case, the digital outputs of the conventional ADC are observed and studied. A sample of a UMTS signal is fed

through an ADC, and the digital outputs are collected. These digital outputs are compared with the original analog sample to understand their behaviour during signal fluctuation. The observations have led to the development of a real-time configuration algorithm for a pipeline ReADC.

Simulation results of a UMTS signal for an ideal case condition demonstrate that the proposed method configures the ADC parameters according to different amplitude variations for each UMTS frame duration. In reN implementation for the pipeline ReADC, the reduction of power is through two mechanisms. Firstly, the power is minimised with a lower $N_{average}$. Secondly, since the dynamic power consumption is dependent on the SA value, the power consumption is decreased with the reduction of SA, and consequently, the total SA value. However, since the power consumption for the pipeline ReADC is dominated by analog modules, especially the comparator, then power consumption is linearly proportional to $N_{average}$.

In addition, the effect of changing the resolution of the ReADC on the BER performance is also analysed, and the value of N_{min} for the ReADC is determined. The relationship between N and BER has enabled the adaptive algorithm to work within the specific requirement of UMTS.

Chapter 6 analyses the impact of parameter settings on the algorithm proposed in Chapter 5. The purpose of the study is to understand how each component of the algorithm contributes to the final solution. Adaptive performance is evaluated according to the number of samples n in the period $T_{measure}$ and the threshold values vt_l. With an increasing window of n, the accuracy of adaptation is increased but with a slower adaptive response as more samples are required. It is vital to assign correct values for the thresholds to avoid mis-operation of the algorithm. Therefore, selecting the right values for the thresholds is vital. From this investigation, it is concluded that to achieve the best performance of the adaptability of the algorithm, the following values of the parameters are suggested:

- The minimum value of n is equivalent to four UMTS time slots ($4 \times T_{slots}$)

- To cover the full-scale adaptive resolution between 6 *bits* and 10 *bits*, vt_l is set to a range of 30% of the signal reduction that includes $vt(max)$ and $vt(min)$ as in the table below:

threshold (vt)	**value**
$vt(max)$	0.467
$vt(min)$	0.463

- To preserve the required BER, the values of N_{min} for different SF settings are summarised below:

SF	N_{min} [**bits**]
8	11
16	9
32	8
64	6

The study has led to a better understanding of the algorithm and the impact of varying each parameter towards the response of the algorithm. Most importantly, this parametric study for reN is critical to choose the relevant settings for specific applications and subsequently, would be able to predict the response beforehand.

Chapter 7 presents an implementation of the proposed algorithm as an adaptive control module for a pipeline ReADC. It is successfully demonstrated that the architecture performs reN with low implementation complexity. The ASIC implementation of the controller adaptively configures the converter to respond to the amplitude variations and more importantly, has provided a low-power solution for the ReADC system. This is achieved at a faster response since the configuration is independent of the baseband processing. To aid the designers, power estimation models for ReADCs are suggested. These include the power consumption of the ACU is compared with the entire ReADC architecture.

The configuration overhead is regarded as the main bottleneck in ReHW implementation. The configuration overhead limits the implementation of reconfigurable technology in low-power applications and hence, must be kept to a minimum. The proposed hardware requires a minimum additional area on top of the main ReADC architecture, where it can quickly perform a 'detects and changes' mechanism. With ASIC implementation of the adaptive control module, real-time configuration with an area overhead less than 11% is achieved relative to the main ReADC modules. Meanwhile, the additional power overhead consumed by the adaptive module is also small, less than 13% compared with the main modules.

Chapter 8 presents an implementation of the proposed SWAM method using a conventional ADC to detect noise or other rapid-signal variations for further signal improvement. This work provides an opportunity to extend the use of the method for future ADC-based applications. Indirectly, the functionality of the ADC can be expanded. A sample of the UMTS signal is superimposed with AWGN where the SNR value is varied. Simulation results show a correlation between the SA and the SNR levels where the SA value of the ADC digital outputs increases with the increment level of noise. Consequently, by monitoring the SA of the ADC output, the noise level that appears in a signal can be estimated and afterwards, the specific adaptive filtering approach could be applied. This is achieved with a fast response and at low-power, as it is fully implemented in ASIC.

It is also found that using binary weighting, a higher DoF is achieved to distinguish different noise levels distinctly, compared with other weighting types. However, the improved performance is achieved at the expense of slightly higher computation time compared with the linear-weighted and odd-number-weighted algorithms.

Finally, an ASIC module for the binary-weighted algorithm is successfully developed. The performance analysis in terms of timing, area, and power is evaluated. At the architecture level, the detection module consumes less than $1.1mW$ of power in $0.18\mu m$ CMOS technology. Similarly, in terms of size, only a 1230

equivalent 2-input NAND gate density is required. The maximum operation frequency of the module is 100 MHz. From the analysis of the results, it is concluded that the module is significantly low in terms of area and power consumption.

9.3 Thesis Achievements

9.3.1 Chapter 4

In this chapter, the following have been presented:

- The minimum requirements for the ADC parameters for the UMTS/FDD mode are successfully derived using mathematical analysis.
- The lowest resolution is 10 *bits* and this considers the design imperfections for practical implementation.
- The minimum sampling rate for the UMTS/FDD mode is 10 Msps.

9.3.2 Chapter 5

In this chapter, the following have been presented:

- The UMTS building blocks for the baseband and the analog front-end have been studied and the UMTS signal has been generated successfully using MATLAB.
- The UMTS framing is analysed in terms of periodicity and time slots.
- The SWAM method is proposed based upon a binary-weighted hamming distance concept from observation of the nature of the ADC output.
- The proposed method is implemented to a conventional 10-*bit* ADC to determine the SA response at different levels of signal amplitudes and time slots.
- It is concluded that the SA has correlations with different levels of UMTS amplitudes and UMTS time slots.
- The performances are evaluated in terms of BER and adaptive response.

Ideally, it is found that the value of N_{min} for the ReADC for the UMTS/FDD mode is 6 *bits* for SF=64.

- The adaptive resolution is determined by computing and comparing the SA value with the threshold values during a measurement period $T_{measure}$. The converter operates with this computed N value for the rest of the frame T_{idle}.

9.3.3 Chapter 6

In this chapter, the following have been presented:

- The study evaluates the parameter space of the adaptive algorithm in terms of observed period $T_{measure}$ and the threshold settings. The results of the SA analysis and average N value are observed.

- For a reliable result, $T_{measure}$ requires at least four UMTS time slots for the SA computation.

- The analysis also found that the threshold setting determines the value of N_{min}, N_{max}, and $N_{average}$ for the entire conversion process. The minimum value must fulfil the BER requirement.

- In addition, at the baseband, SF is important in data protection where SF=64 gives better immunity against noise than a lower value. Consequently, a high SF value implies a low N_{min} value, and the requirement for higher resolution ADC can be relaxed.

9.3.4 Chapter 7

In this chapter, the following have been presented:

- This chapter focuses on ASIC design, simulation, and implementation of an ACU module for a pipeline ReADC. The ACU module is the realisation of the SWAM algorithm.

- Power estimation models for ReADCs at the system level are presented as

reconfiguration performance metrics to aid system designers.

- The ASIC performance result exhibits small area complexity with gate density around 3090 equivalent 2-input NAND gates in a $0.18\mu m/1.8V$ CMOS process.

- The power consumption of the ACU is only $1907\mu W$.

9.3.5 Chapter 8

In this chapter, the following have been presented:

- Concept, simulation, and implementation of the noise correlation analysis and detection circuit for ADC-based electronic devices using the SWAM method are presented. The SA of the digital output of the ADC is used to differentiate between noisy and noiseless signals. In the presence of noise, the SA of noisy signals is larger than that of noiseless signals as the MATLAB simulation results verify. The results show that the SA of digital outputs varies with different SNR values, where a high SNR value leads to a low SA and vice versa.

- By introducing a weighting element to the SWAM algorithm, a better range of detection can be achieved. The binary-weighted algorithm has increased the DoF for decision making significantly. The weighted algorithm increases the range of the total switching activity value by 155% compared with linear- and odd number-weighted implementations.

- The ASIC performance result shows that the NDU module occupies a small footprint, around 1230 2-input NAND equivalent gate density.

- The hardware is also low power, consuming just under $1100\mu W$ in $0.18\mu m/1.8V$ CMOS technology.

9.3.6 Key Research Contributions

In summary, the key contributions of the thesis are:

- A derivation of minimum requirements for an ADC based on a target application (Chapter 4).

- An analysis of UMTS signals according to periodicity and framing structure (Chapter 5).

- A proposed detection method based upon observation of the digital output of a conventional ADC. The SWAM method is developed using a binary-weighted hamming distance concept that monitors changes of the digital output (Chapter 5).

- Development of an adaptive algorithm for ReADC based on the SWAM method. The ReADC is able to adapt its resolution depending on the variation in the signal amplitude (Chapter 5).

- Performance analysis of the adaptive algorithm on different parameter settings to optimise the algorithm operation (Chapter 6).

- A technique to measure the area and power overheads for both reN and reFs implementations (Chapter 7).

- Development of a low-power adaptive control module for the pipeline ReADC to vary its resolution adaptively (Chapter 7).

- Correlation analysis between the SA and different SNR levels and consequently, the development of a noise detection circuit from the analysis results (Chapter 8).

- Implementation of the proposed SWAM method in ADC-based system to adapt to the signal change and to improve the signal quality (Chapter 8).

9.4 Thesis Conclusion

Power consumption is an important issue in electronics and a concern for the environmentalists. As functionalities of electronic devices grow, their power consumption increases. In wireless communication, the role of reconfigurability and flexibility architecture has become increasingly important as a technology that supports multiple applications and multi-mode operations. However, issues associated with reconfigurability, such as power overhead, have to be scrutinised for successful implementation. This thesis aims to resolve the issues associated with power consumption and reconfigurability for ReADCs, simultaneously.

In this thesis, an adaptive technique for ReADCs at algorithmic and architectural levels is presented. In particular, real-time configuration is implemented to a pipeline ReADC targeting low-power wireless devices. This is achieved by exploiting the SA of the ReADC digital output. Consequently, a method based on the SA measurement, the SWAM algorithm, is proposed. The thesis also investigates the impact of varying parameter settings of the algorithm on the adaptability, response, and diversity of the solutions. Threshold values vt_l, number of samples n in $T_{measure}$ duration, and ReADC wordlength for the SA calculation N_{xt} are the factors that contributes to the quality of adaptive performance.

The implementation of the SWAM algorithm as a low-power adaptive controller module, known as the ACU, has enabled the ReADC to monitor, evaluate, and respond to the magnitude variations of the UMTS signal and configures its resolution adaptively according to the current changes. The reconfiguration does not require the baseband processing, and hence, requires short response time. Real-time adjustment of the resolution, whose range is set by N_{min} and N_{max} that must fulfil the BER and power requirements, allows the power consumption of the ReADC to be scaled according to the signal variations, as opposed to the conventional ADC that is fixed throughout its operation.

The power consumed by the ACU is significantly lower as compared with the existing control mechanisms that have been implemented using DSPs. The maxi-

mum power consumption of the ReADC system with the integrated ACU module is kept close to the conventional 10-*bit* ADC when operating in full resolution mode, while approximately up to 40% of power is saved in minimum resolution mode. This power reduction is achieved via the use of the SWAM method as the ACU module that is realised in ASIC, ensuring the implementation with low reconfigurable overheads and a fast response. Consequently, the maximum power consumption of the ReADC is maintained at the level of the ADC while additional reconfiguration features are introduced.

The SWAM algorithm can also be utilised in conjunction with the conventional ADC to detect different levels of noise strength in the signal. Such a feature can be useful for signal enhancement and data recovery. Incorporating the NDU in the conventional ADC requires less than 2% of an additional area compared with the existing ADC unit of the same CMOS process technology. The amount of power required by the NDU is only 4.5% by the same comparison.

In this work, the low-power strategy exploits the digital section of the ReADC. Since the total power of the ReADC is mainly dominated by the analog part, by manipulating the digital part, the power consumption of the converter is maintained at the improved features. The reconfigurability aspect of the ReADC is enhanced with the adaptive feature controlled by the ReADC itself. Consequently, the requirement for external control signals from the baseband to configure the ReADC becomes unnecessary.

In short, the use of the SWAM algorithm as a sensing mechanism demonstrates that it could be used either to enhance the previous works as an adaptive controller for ReADCs, or to expand the functionality of existing ADCs as preliminary noise detection. This is indicated by the implementation results of the SWAM method to the ADC that reveal reconfigurabilty, low power consumption, and a small footprint at higher functionality have been achieved. Therefore, this method has shown potential for future adaptive and reconfigurable hardware with low-power approach.

9.5 Future Work

This thesis has endeavoured to provide rigorous investigation into the field of ReHW and open doors for future research and development.

9.5.1 Extending Range of Analysis

This thesis indicates three areas that require further investigation to improve the solutions.

Firstly, the analysis of thresholds only considers the reduction in half- and quarter power, which implies the reduction in signal amplitude up to 30% and 50%, respectively. A wider range of analyses of amplitude changes may also help yield a better adaptation.

Secondly, binary-weighted coefficients have improved the solution of the adaptive algorithm, but at the expense of higher computation than the linear-weighted and odd-number-weighted coefficients. By analysing different weighting sequences, such as Fibonacci and square-number sequences towards the DoF and the effectiveness of the solution depending upon applications, other suitable weighted coefficients may be implemented.

In the design and implementation, the entire ADC wordlength is used for SA computation and is of 10-*bit* length. Through analysing different values of N_{xt}, the ADC wordlength for SA computation, the computational complexity could be reduced but still preserve the required results.

9.5.2 Power Characterisation

The work in this thesis focuses on the ASIC control mechanism of the ReADC and has the following limitations:

No exact comparison with other work on pipeline ReADCs is available that used different adaptive techniques because the controller mechanisms of the existing ReADCs are implemented using DSP.

In addition, the main pipeline ReADC architecture, including the digital encoding unit, is referred to other available published articles. Thus, the power consumption of the adaptive control unit is evaluated relative to the existing ReADC architectures using power estimation model.

Currently, the actual ReADC performance in terms of speed and power consumption cannot be evaluated since this work focus entirely on the digital controller part. This can be resolved in the future when analog circuitry of pipeline ReADC is developed and simulated with the proposed solution both at architectural and circuit level designs. Therefore, the ASIC performance of the ReADC in terms of response, speed, and power could be evaluated by combining the pipeline ReADC with the proposed energy efficient ACU. This will enable the performance of the entire system to be evaluated accurately and the total power consumption of ReADC to be measured precisely.

9.5.3 Design Improvement

In this work, the digital encoding of the ADC is truncated when operating in lower resolution mode. To improve the accuracy of the digital output, a digital encoder unit of ReADC could be reconfigured with a suitable decoding algorithm scheme.

Furthermore, to further optimise both power and the area of the ACU, the hardware implementation of the algorithm could also be improved by using better Verilog coding. At the moment, the developed hardware performs the algorithm successfully without design, area, and power optimisation.

Similarly, the design only considers reN implementation. The design could be expanded with the implementation of reFs. Specifically for the wireless implementation, since the frequency is fixed, the implementation of reFs is unnecessary. However, for sensor applications, where a signal of different frequencies can be detected, reFs implementation could reduce the power consumption since the converter could be set to operate at higher frequencies only when necessary.

For the NDU implementation, the module could be improved by introducing the look-up-table (LUT) in the memory for different levels of noise, as referred to the simulation results. Although the cell size will be larger than the implemented NDU, different SNR and noise strength could be classified distinctively.

9.5.4 Widening Scope of Applications

The proposed SWAM method and the adaptive algorithm could be extended in several ways. First, an evaluation of the effectiveness of the adaptive control unit in other typical ADC architectures, such as flash and SAR ADCs, should be considered. This is to determine the use of the SWAM algorithm in broader aspects of ReADC implementation. Secondly, a wider variation of signals and application could be considered. Sensors and other low frequency signal applications can be used to determine the effectiveness of the algorithm in the wider scope. Finally, the proposed SWAM method using ADC is promising for controlling electronic circuits. Detailed investigation will enable further exploration of the use of the SWAM method in signal analysis and digital control for other areas and architectures.

9.6 Final Comment

ReHW is a new research area that is important for modern consumer electronics and wireless devices. It holds many advantages in terms of flexibility, reusability, adaptability, and efficient power utilisation, which ultimately will lead to green device technology. With the widespread use of mixed-mode applications, research in ReHW has become vital. As most digital devices require ADC to digitise analog signals for further processing, research in ReADC is equally essential. While the advantages of ReADC are evident, the main issues in real-time configurable hardware have been dealing with detection mechanisms and reconfigurable overheads. This thesis has proposed several solutions to these reconfigurability problems.

Research in ReADC has become more prominent since the last decade. In the next ten years, more sophisticated techniques and applications of ReADCs for consumer electronics will likely be developed. Increasing demand from consumers for more powerful devices with better features and low-power consumption will force manufacturers to produce improved configuration engines. Thus, pressure for efficient adaptive algorithms and architectures for ReADCs could be even greater in the near future.

Publications

1. **Z. Razak**, A. Erdogan, and T. Arslan, *"Nyquist-rate Analog-to-Digital Converter Specification for Zero-IF UMTS Receiver"*, IEEE International Symposium on Circuits and Systems (ISCAS'08), USA, pp. 2338 – 2341, May. 2008.

2. **Z. Razak**, A. Erdogan, and T. Arslan, *"Analog to Digital Converter Specification for UMTS/FDD Receiver Applications"*, 4th IEEE International Symposium on Electronic Design, Test and Applications (DELTA'08), Hong Kong, pp. 446 – 449, Jan. 2008.

3. **Z. Razak**, A. Erdogan, and T. Arslan, *"An Adaptive Algorithm for Reconfigurable Analog-to-Digital Converters"*, NASA/ESA Conference on Adaptive Hardware and Systems (AHS'10), pp. 250 – 257, USA, Jun. 2010.

4. **Z. Razak**, A. Erdogan, and T. Arslan, *"ASIC Design of an Adaptive Control Unit for Reconfigurable Analog-to-Digital Converters"*, IEEE Computer Society Annual Symposium on VLSI (ISVLSI'10), Greece, pp. 179 – 184, Jul. 2010.

5. **Z. Razak**, A. Erdogan, and T. Arslan, *"Low Power Noise Detection Circuit Utilizing Switching Activity Measurement Method"*, Electronic Chips & Systems Design Initiative (ECSI) Conference on Design and Architectures for Signal and Image Processing (DASIP'10), UK, pp. 258 – 264, Oct. 2010.

Acronyms and Abbreviations

3G	3rd generation of cellular standards
3GPP	3rd generation partnership project
4G	4th generation of cellular standards
AC	alternating current
ACS	adjacent channel selectivity
ACU	adaptive control unit
ADC	analog-to-digital converter
AGC	automatic gain control
AHW	adaptive hardware
ALE	adaptive line enhancer
ASIC	application-specific integrated circuit
AWGN	additive white gaussian noise
AU	analog unit
BER	bit-to-error rate
BPF	band-pass filter
CE	configurable element

CDMA code division multiple access

CLB clustered logic block

CMOS complementary metal oxide semiconductor

CRC cyclic redundancy check

CT continuous time

CU configuration unit

CW continuous wave

DAC digital-to-analog converter

DC direct current

DFF D flip-flop

DoF degree of freedom

DPCH dedicated physical channel

DPCCH dedicated physical control channel

DPDCH dedicated physical data channel

DR dynamic range

DS-CDMA direct sequence CDMA

DSM deep submicron

DSP digital signal processor

DU digital unit

EDGE enhanced data rates for GSM evolution

FDD frequency division duplex

FFT	fast fourier transform
FOM	figure of merit
FPAA	field programmable analog array
FPGA	field programmable gate arrays
FPTA	field programmable transistor array
FS	full-scale
GA	genetic algorithm
GMM	global multimedia mobility
GP	genetic programming
GPS	global positioning system
GSM	global system for mobile communications
HDTV	high-definition television
IC	integrated circuit
IF	intermediate frequency
IMT-2000	international mobile telecommunication 2000
ITU	international telecommunication union
LNA	low-noise amplifier
LO	local oscillator
LSB	least significant bit
LUT	look-up-table
MDAC	multipying-DAC

ME memory elements

MS mobile station

MSB most significant bit

NDU noise detection unit

NMOS n-channel MOS

NRE non-recurring engineering cost

OSR oversampling ratio

OVSF orthogonal variable spreading factor

PA power amplifier

PAU programmable analog unit

PC personal computer

PDU programmable digital unit

PLL phase lock loop

PMOS p-channel MOS

PN pseudo-noise

POR power-on reset

PNR place and route

PSD power spectral density

PSRR power supply rejection ratio

PWM pulse width modulation

QPSK quadrature phase shift keying

QoS	quality of service
reN	reconfigurable resolution
reFs	reconfigurable sampling-rate
RC	reconfigurable computing
ReADC	reconfigurable ADC
ReHW	reconfigurable hardware
RF	radio frequency
RFID	RF identification
RND	research and development
RRC	root-raised cosine
RTL	register transfer level
Rx	receiver
SAR	successive approximation
SAIF	switching activity interchange format
SAW	surface acoustic wave
SBCS	self-biased current source
SC	switched capacitor
SDR	software-defined radio
SDTV	standard definition television
SF	spreading factor
SH	sample-and-hold

SiP system-in-a-package

SNR signal-to-noise ratio

SoC system-on-a-chip

SWAM switching activity measurement

TDD time division duplex

TI time-interleaved

TTI transmission time interval

Tx transmitter

UE user equipment

UMTS universal mobile telecommunication system

UMTS/FDD UMTS frequency division duplex

UMTS/TDD UMTS time division duplex

UWB ultra wideband

VGA variable-gain amplifier

WCDMA wideband CDMA

WiFi wireless fidelity

WiMAX worldwide interoperability for microwave access

WLAN wireless LAN

XOR XOR gate

ZERO-IF homodyne receiver

Nomenclature

α_i	i-th SA value
$\alpha_{0\to1}$	switching activity at nodes
β	binary coefficient
δ	a small change in the value
κ	power coefficient depending on circuit elements
$\langle S_t \rangle$	mean value of total switching activity
ω_c	carrier angular frequency
ω_m	signal angular frequency
$\Sigma\Delta$	sigma-delta
A_v	amplifier gain
A_{ACU}	ACU area
A_{ADC}	ADC area
A_{NDU}	NDU area
A_{ReADC}	ReADC area
B_i	ADC output at i-th sample
b_j	*bit* at $j-$th position

bit	unit of ADC resolution
BW	bandwidth
C	capacitance
C_{load}	load capacitance at gate
CF	crest factor
$cntrT0\_end$	duration of $T_{measure}$
$cntrT\_end$	end of counter for duration of $T_{measure}$
$DPCH\_E_c$	reference sensitivity level
E_b	energy per information bit
E_b/N_0	bit energy per noise density
$endConv$	end of ADC conversion
$ENOB$	effective number of bit
f	clock frequency
f_c	carrier frequency
f_s	sampling frequency
$f_s(min)$	minimum sampling frequency
f_{sig}	signal frequency
G_c	coding gain
G_p	processing gain
G_s	spreading gain
GHz	gigahertz

hex	hexadecimal
I	in-phase signal
i	$i-th$ sample
I_C	channel current
I_{bias}	bias current
$I_{leakage}$	leakage current
I_{SC}	short-circuit current
I_{total}	total current
j	$j-th$ position of the ADC wordlength
k	Boltzmann's constant
k_α	analog circuit coefficient
l	$l-th$ threshold level
L_{min}	minimum CMOS channel length
M	total pipeline stage
m	active operational stages of ReADC
mem	memory cell
MHz	megahertz
N	resolution
n	number of sample
N_0	noise spectral density
$N_{average}$	average resolution

N_{max}	maximum resolution
n_{max}	maximum acceptable noise
N_{min}	minimum resolution
N_{new}	new resolution value
N_{xt}	ADC resolution for calculating switching activity
NF	noise figure
$noise\_detect$	noise detection flag
P_{ACU}	ACU power
P_{ADC}	ADC power
P_{ave}	mean value of power
P_{block}	single-stage pipeline power
$P_{comparator}$	comparator power
P_{DEC}	digital decoder circuit power
P_{DSP}	DSP power
P_{dyn}	dynamic power
$P_{leakage}$	leakage power
P_{NDU}	NDU power
P_{PAU}	PAU power
P_{PDU}	PDU power
P_{peak}	peak amplitude of instantaneous power
P_{ReADC}	ReADC power

P_{SC}	short-circuit power
P_{SH}	SH circuit power
$P_{switching}$	switching power
P_{total}	total power consumption
Q	quadrature signal
q	total number of frame
Q_{ε}	quantisation error
Q_c	amount of charge
$r(t)$	received signal
R_b	bit rate
R_c	chip rate
R_S	source resistor
R_s	symbol rate
$readc$	reN flag
$s(t)$	transmitted signal
S_t	total switching activity value
$S_t(accum.\ max.)$	maximum accumulative value of switching activity
$S_t(accum.)$	accumulative value of switching activity
$S_t(avg)$	average value of switching activity
$S_t(MAX)$	maximum theoretical value of switching activity
$S_t(max)$	maximum measured value of switching activity

$S_t(min)$	minimum measured value of switching activity
$S_t(norm)$	normalised value of switching activity
S_w	switching activity factor
$S_w.C_{load}$	switched capacitance
SA	switching activity
SA_{ref}	reference value of switching activity
SNR_i	SNR at input
SNR_o	SNR at output
T	observed interval
T_K	absolute temperature in unit Kelvin
t_s	sampling interval
T_{chip}	chip duration
T_{frame}	frame duration
T_{idle}	interval of no measurement
$T_{max\ changes}$	maximum SA changes in an observed interval
$T_{measure}$	measurement interval
T_{slot}	duration of a UMTS slot
$T_{super\ frame}$	superframe duration
T_s	duration of sampling-rate
T_{win}	total observed interval
V_S	source voltage

V_{dd}	supply voltage
V_{eff}	effective voltage
V_{FS}	full-scale voltage
V_{GS}	gate-source voltage
V_{LSB}	LSB voltage
V_{pp}	peak-to-peak voltage
V_{rms}	root-mean-square voltage
V_{swing}	swing voltage
V_{TH}	threshold voltage
$vt(max)$	maximum threshold level
$vt(min)$	minimum threshold level
vt_l	threshold level
$^1/_f$	flicker noise

Bibliography

[1] T. S. Hall, *Field-Programmable Analog Arrays: A Floating-Gate Approach*. PhD Dissertation, School of Electrical and Computer Engineering, Georgia Institute of Technology, Georgia Institute of Technology, Atlanta, Georgia, US, Jul 2004.

[2] E. K. F. Lee and W. L. Hui, "A Novel Switched-Capacitor Based Field-Programmable Analog Array Architecture," *Analog Integrated Circuits Signal Processing Journal - Special Issue on Field Programmable Analog Arrays*, vol. 17, pp. 35–50, Sep 1998.

[3] F. Francesconi and F. Maloberti, "A Low Power Logarithmic A/D Converter," in *Proceedings of IEEE International Symposium on Circuit and System (ISCAS'96)*, vol. 1, pp. 473–476, May 1996.

[4] D. A. Johns and K. Martin, *Analog Integrated Circuit Design*. John Wiley & Sons, 1997.

[5] B. Murmann, "A/D Converter Trends: Power Dissipation, Scaling and Digitally Assisted Architectures," in *IEEE Custom Integrated Circuits Conference (CICC'08)*, pp. 105–112, Sep 2008.

[6] J. Mitola III, "Software Radios: Survey, Critical Evaluation and Future Directions," *IEEE Aerospace and Electronic Systems Magazine*, vol. 8, pp. 25–36, Apr 1993.

[7] S. Khawam, I. Nousias, M. Milward, Y. Yi, M. Muir, and T. Arslan, "The Reconfigurable Instruction Cell Array," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, pp. 75 – 85, Jan 2008.

[8] A. Stoica, "Toward Evolvable Hardware Chips: Experiments with a Programmable Transistor Array," in *Proceedings of the Seventh International Conference on Microelectronics for Neural, Fuzzy and Bio-Inspired Systems (MicroNeuro'99)*, pp. 156 –162, Apr 1999.

[9] D. R. D'Mello and P. G. Gulak, "Design Approaches to Field-Programmable Analog Integrated Circuits," *Analog Integrated Circuits Signal Processing*, vol. 17, pp. 7 – 34, Sep 1998.

[10] K. Compton and S. Hauck, "Reconfigurable Computing: A Survey of Systems and Software," *ACM Computer Surveys*, vol. 34, pp. 171 – 210, Jun 2002.

[11] A. DeHon and J. Wawrzynek, "Reconfigurable Computing: What, Why, and Implications for Design Automation," in *Proceedings of the 36th annual ACM/IEEE Design Automation Conference (DAC'99)*, pp. 610–615, Jun 1999.

[12] A. Stoica, "Evolvable Hardware: From On-Chip Circuit Synthesis to Evolvable Space Systems," in *IEEE International Symposium on Multiple-Valued Logic (ISMVL'00)*, pp. 161 – 169, Nov 2000.

[13] J. R. Koza, F. H. Bennett III, D. Andre, and M. A. Keane, "Automated WYWIWYG Design of Both the Topology and Component Values of Analog Electrical Circuits Using Genetic Programming," in *Proceeding of Genetic Programming Conference*, (Stanford University, USA), pp. 28–31, MIT Press, 1996.

[14] E. K. Lee and P. Gulak, "A CMOS Field-Programmable Analog Array," in *38th IEEE International Solid-State Circuits Conference (ISSCC'91). Digest of Technical Papers.*, pp. 186–314, Feb 1991.

[15] J. Langeheine, J. Becker, S. Fölling, and K. Meier, "Initial Studies of a New VLSI Field Programmable Transistor Array," in *Proceedings 4th International Conference on Evolvable Systems: From Biology to Hardware*, vol. 2210 of *Lecture Notes in Computer Science*, pp. 62–73, Springer Verlag, 2001.

[16] T. Higuchi, Y. Liu, and X. Yao, *Evolvable Hardware*. Springer, 2006.

[17] P. Garcia, K. Compton, M. Schulte, E. Blem, and W. Fu, "An Overview of Reconfigurable Hardware in Embedded Systems," *EURASIP Journal of Embedded Systems*, pp. 1–19, Jun 2006.

[18] T. van As and S. Krijgsman, "Reconfigurable Architectures: A Survey of Design and Implementation Methods," *Faculty of Electrical Engineering, Mathematics & Computer Science, Delft University of Technology*, Jul 2007.

[19] T. Todman, G. Constantinides, S. Wilton, O. Mencer, W. Luk, and P. Cheung, "Reconfigurable Computing: Architectures and Design Methods," *IEE Proceedings of Computers and Digital Techniques*, vol. 152, pp. 193 – 207, Mar 2005.

[20] H. Sakanashi, M. Iwata, D. Keymulen, M. Murakawa, I. Kajitani, M. Tanaka, and T. Higuchi, "Evolvable Hardware Chips and Their Applications," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC'99)*, vol. 5, pp. 559 – 564, Oct 1999.

[21] J. C. Gallagher, S. Vigraham, and G. Kramer, "A Family of Compact Genetic Algorithms for Instrinsic Evolvable Hardware," *IEEE Transactions on Evolutionary Computation*, vol. 8, pp. 111 – 126, Apr 2004.

[22] M. Anderson and J. Yuan, "A Novel Reconfigurable Pipelined A/D Conversion Technique for Multistandard Wideband Receivers," in *Proceedings of the IASTED International Conference on Circuits, Signals, and Systems (CSS'04)*, (Clearwater Beach, FL, United States), pp. 199–204, Dec 2004.

[23] J. M. D. Pereira, O. Postolache, and P. S. Girao, "A Digitally Programmable A/D Converter for Smart Sensors Applications," in *Proceedings of the IEEE Instrumentation and Measurement Technology Conference (IMTC'05)*, vol. 1, pp. 156–159, May 2005.

[24] J. Mitola, "Technical Challenges in the Globalization of Software Radio," *IEEE Communications Magazine*, vol. 37, pp. 84–89, Feb 1999.

[25] D. Chester, "Digital IF Filter Technology for 3G Systems: An Introduction," *IEEE Communications Magazine*, vol. 37, pp. 102–107, Feb 1999.

[26] J. M. P. Cardoso and P. C. Diniz, *Compilation Techniques for Reconfigurable Architectures*. Springer, 2009.

[27] A. Vigna, *Design of a D/A Converter for Reconfigurable Mobile Terminals*. PhD Dissertation, University of Pavia, Italy, 2005.

[28] ITU, "International Telecommunication Union: Mobile Telephony Statistics." `http://www.itu.int/ITU-D/ict/statistics/`, Jun 2011.

[29] H. Holma and A. Toskala, *WCDMA for UMTS*. John Wiley & Sons, 2000.

[30] A. Springer and R. Weigel, *The UMTS Physical Layer: Basics, Standard, and Frontend Matters*. Springer-Verlag, 2001.

[31] B. Walke, *Mobilfunknetze und ihre Protokolle Band 1: Grundlagen, GSM, UMTS, und andere zellulare Mobilfunknetze*. Teubner Verlag, 2000.

[32] A. Springer, L. Maurer, and R. Weigel, "RF System Concepts for Highly Integrated RFICs for W-CDMA Mobile Radio Terminals," *IEEE Transactions on Microwave Theory Technology*, vol. 50, pp. 254 – 267, Jan 2002.

[33] R. Ormondroyd and J. Maxey, "Performance of Low-Rate Orthogonal Convolutional Codes in DS-CDMA Applications," *IEEE Transactions on Vehicular Technology*, vol. 46, pp. 320 – 328, May 1997.

[34] A. N. Akansu and R. Poluri, "Walsh-Like Nonlinear Phase Orthogonal Codes for Direct Sequence CDMA Communications," *IEEE Transactions on Signal Processing*, vol. 55, pp. 3800 – 3806, Jul 2007.

[35] H. F. Pretl, *Design of a Zero-IF Receiver for UMTS*. PhD Dissertation, Institute for Telecommunication Engineering and Information Technology, Johannes Kepler University, Linz, Austria, Jul 2001.

[36] A. Springer and R. Weigel, *UMTS: The Universal Mobile Telecommunications Systems*. Springer-Verlag, 2002.

[37] P. E. Allen and D. R. Holberg, *CMOS Analog Circuit Design*. Prentice Hall, 1987.

[38] W. Kester, *Data Conversion Handbook (Analog Devices)*. Newnes, 2004.

[39] Y.-T. Wang and B. Razavi, "An 8-bit 150-MHz CMOS A/D Converter," *IEEE Journal of Solid-State Circuits*, vol. 35, pp. 308 – 317, Mar 2000.

[40] I. Ahmed, *Pipelined ADC Design and Enhancement Techniques*. Springer, 2010.

[41] P. Gray and D. Hodges, "All-MOS Analog Digital Conversion Techniques," *IEEE Transactions on Circuits and Systems*, vol. 25, no. 7, pp. 482–489, Jul 1978.

[42] T.-S. Lee and K.-R. Hsiao, "A Low-Voltage Fully Differential CMOS High-Speed Sample-and-Hold Circuit," in *Proceedings of the 44th IEEE 2001 Midwest Symposium on Circuits and Systems (MWSCAS'01)*, vol. 1, pp. 235 – 238, Aug 2001.

[43] P. R. Gray and R. G. Meyer, "MOS Operational Amplifier Design – A Tutorial Overview," *IEEE Journal of Solid-State Circuits*, vol. 17, pp. 969–982, Dec 1982.

[44] D. J. Allstot, "Precision Variable-Supply CMOS Comparator," *IEEE Journal of Solid-State Circuits*, vol. SC-17, pp. 1080 – 1087, Dec 1982.

[45] B. Razavi, *Principles of Data Conversion System Design*. IEEE Press, 1995.

[46] B. McCarroll, C. Sodini, and H.-S. Lee, "A High-Speed CMOS Comparator for Use in an ADC," *IEEE Journal of Solid-State Circuits*, vol. 23, pp. 159–165, Feb 1988.

[47] W.-T. Lin and T.-H. Kuo, "A Compact Dynamic-Performance-Improved Current-Steering DAC With Random Rotation-Based Binary-Weighted Selection," *IEEE Journal of Solid-State Circuits*, vol. 47, pp. 444 – 453, Feb 2012.

[48] NI, "National Semiconductor: ADC Encoding Schemes." `http://www.national.com/AU/design/courses/121/adc03/01adc03.htm`.

[49] Y.-T. Lai and C.-N. Yeh, "A Folding Technique for Reducing Circuit Complexity of Flash ADC Decoders," *Analog Integrated Circuits and Signal Processing*, vol. 63, pp. 339 – 348, May 2010.

[50] U.-K. Moon and B.-S. Song, "Background Digital Calibration Techniques for Pipelined ADCs," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, pp. 102 – 109, Feb 1997.

[51] M. Taherzadeh-Sani and A. Hamoui, "Digital Background Calibration of Capacitor-Mismatch Errors in Pipelined ADCs," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, pp. 966 – 970, Sep 2006.

[52] H.-C. Hong and G.-M. Lee, "A 65-fJ/Conversion-Step 0.9-V 200-kS/s Rail-to-Rail 8-bit Successive Approximation ADC," *IEEE Journal of Solid-State Circuits*, vol. 42, pp. 2161 – 2168, Oct 2007.

[53] W. Liu and Y. Chiu, "Background Digital Calibration of Successive Approximation ADC with Adaptive Equalisation," *Electronics Letters*, vol. 45, pp. 456–458, Apr 2009.

[54] B. Boser and B. Wooley, "The Design of $\Sigma\Delta$ Modulation Analog-to-Digital Converters," *IEEE Journal of Solid-State Circuits*, vol. 23, pp. 1298 – 1308, Dec 1988.

[55] C. Jun, Z. Changlu, and X. Guanhuai, "A Fourth-Order 18-b Delta-Sigma A/D Converter," in *Conference on High Density Microsystem Design and Packaging and Component Failure Analysis*, pp. 1 – 4, Jun 2005.

[56] B.-G. Lee, B.-M. Min, G. Manganaro, and J. Valvano, "A 14-b 100-MS/s Pipelined ADC With a Merged SHA and First MDAC," *IEEE Journal of Solid-State Circuits*, vol. 43, pp. 2613 – 2619, Dec 2008.

[57] I. Ahmed and D. Johns, "A High Bandwidth Power Scalable Sub-Sampling 10-Bit Pipelined ADC With Embedded Sample and Hold," *IEEE Journal of Solid-State Circuits*, vol. 43, pp. 1638–1647, Jul 2008.

[58] H. Iwai, "CMOS Technology Future," in *Proceedings of the 5th IEEE International Caracas Conference on Devices, Circuits and Systems*, vol. 1, pp. 179 – 182, Nov 2004.

[59] B. Razavi, "CMOS Technology Characterization for Analog and RF Design," *IEEE Journal of Solid-State Circuits*, vol. 34, pp. 268–276, Mar 1999.

[60] M. Degrauwe, J. Rijmenants, E. Vittoz, and H. De Man, "Adaptive Biasing CMOS Amplifiers," *IEEE Journal of Solid-State Circuits*, vol. 17, pp. 522–528, Jun 1982.

[61] E. Camacho-Galeano, C. Galup-Montoro, and M. Schneider, "An Ultra-Low-Power Self-Biased Current Reference," in *Proceedings of 17th Symposium on Integrated Circuits and Systems Design (SBCCI'04)*, (New York, USA), pp. 147–150, Sep 2004.

[62] P. Furth and H. Ommani, "Low-Voltage Highly-Linear Transconductor Design in Subthreshold CMOS," in *Proceedings of the 40th Midwest Symposium on Circuits and Systems (MWCAS'97)*, vol. 1, pp. 156 – 159, Aug 1997.

[63] S. DeWeerth, G. Patel, and M. Simoni, "Variable Linear-Range Subthreshold OTA," *Electronics Letters*, vol. 33, pp. 1309–1311, Jul 1997.

[64] D. Schmitt and T. Fiez, "A Low Voltage CMOS Current Source," *Proceedings of International Symposium on Low Power Electronics and Design*, pp. 110 – 113, Aug 1997.

[65] M. Steyaert, A. H. M. v. Roermund, and J. H. Huijsing, *Analog Circuit Design - RF Circuits: Wide band, Front-Ends, DAC's, Design Methodology and Verification for RF and Mixed-Signal Systems, Low-Power and Low Voltage*. Springer, 2006.

[66] P. B. Kenington and L. Astier, "Power Consumption of A/D Converters for Software Radio Applications," *IEEE Transactions on Vehicular Technology*, vol. 49, pp. 643 – 650, Mar 2000.

[67] E. Lauwers and G. Gielen, "Power Estimation Methods for Analog Circuits for Architectural Exploration of Integrated Systems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 10, pp. 155–162, Apr 2002.

[68] P. Wambacq, G. Gieles, and J. Gerrits, *Low-Power Design Techniques and CAD Tools for Analog and RF Integrated Circuits*. Kluwer Academic Publishers, 2003.

[69] Z. Huang and P. Zhong, "An Architectural Power Estimator for Analog-to-Digital Converters," in *Proceedings of IEEE International Conference on Computer Design: VLSI in Computers and Processors (ICCD'04)*, pp. 397–400, Oct 2004.

[70] E. K. F. Lee, "Reconfigurable Pipelined Data Converter Architecture," in *Proceedings of the 39th Midwest Symposium on Circuits and Systems (MWSCAS'96)*, vol. 1, Aug 1996.

[71] S. Mortezapour and E. Lee, "A Reconfigurable Pipelined Data Converter," in *The 2001 IEEE International Symposium on Circuits and Systems (ISCAS'01)*, vol. 4, pp. 314 – 317, May 2001.

[72] D. Angele, M. Stein, J. Kauffman, M. Ortmanns, and J. Becker, "A Reconfigurable Continuous-Time $\Sigma\Delta$-ADC using a Digitally Programmable $G_m - C$ Array," in *IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS'12)*, pp. 810–813, Aug 2012.

[73] C. Zierhofer, "Adaptive $\Sigma\Delta$ Modulation with One-bit Quantization," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, pp. 408–415, May 2000.

[74] L. Hui and M. Hassoun, "A 9-b 40-MSample/s Reconfigurable Pipeline Analog-to-Digital Converter," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 49, pp. 449–456, Jul 2002.

[75] G. Rajashekar and M. Bhat, "Design of Resolution Adaptive TIQ Flash ADC using AMS 0.35μm Technology," in *International Conference on Electronic Design (ICED'08)*, pp. 1–6, Dec 2008.

[76] X. Gang and J. Yuan, "A Programmable Analog-to-Digital Converter," in *Proceedings of the 2004 International Symposium on Circuits and Systems (ISCAS'04)*, vol. 1, pp. I–585 – I–588, May 2004.

[77] C. E. Onete, "Reconfigurable Flash A/D Converters," in *IEEE International SOC Conference (SOCC'08)*, pp. 323–326, Sep 2008.

[78] K. Gulati and H.-S. Lee, "A Low-Power Reconfigurable Analog-to-Digital Converter," *IEEE Journal of Solid-State Circuits*, vol. 36, pp. 1900 – 1911, Dec 2001.

[79] A. Zanikopoulos, P. Harpe, H. Hegt, and A. van Roermund, "A Flexible ADC Approach for Mixed-Signal SoC Platforms," in *IEEE International Symposium on Circuits and Systems (ISCAS'05)*, vol. 5, (Kobe, Japan), pp. 4839–4842, May 2005.

[80] Y.-J. Kim, H.-C. Choi, S.-W. Yoo, S.-H. Lee, D.-Y. Chung, K.-H. Moon, H.-J. Park, and J.-W. Kim, "A Re-configurable 0.5V to 1.2V, 10MS/s to 100MS/s, Low-Power 10b 0.13*um* CMOS Pipeline ADC," in *2007 IEEE Custom Integrated Circuits Conference (CICC'07)*, pp. 185–188, Sep 2007.

[81] P. Setty, J. Barner, J. Plany, H. Burger, and J. Sonntag, "A 5.75 bit 350 M sample/s or 6.75 b 150 M sample/s Reconfigurable Flash ADC for a PRML Read Channel," in *IEEE 45th International Solid-State and Circuits Conference (ISSCC'98)*, pp. 148–149, Feb 1998.

[82] S. O'Driscoll, K. Shenoy, and T. Meng, "Adaptive Resolution ADC Array for an Implantable Neural Sensor," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 5, pp. 120–130, Apr 2011.

[83] R. Sekimoto, A. Shikata, T. Kuroda, and H. Ishikuro, "A 40*nm* 50S/s - 8MS/s Ultra Low Voltage SAR ADC with Timing Optimized Asynchronous Clock Generator," in *2011 Proceedings of the 41st European Solid-State Circuits Conference (ESSCIRC'11)*, pp. 471–474, Sep 2011.

[84] M. Yip and A. Chandrakasan, "A Resolution-Reconfigurable 5-to-10-Bit 0.4-to-1 V Power Scalable SAR ADC for Sensor Applications," *IEEE Journal of Solid-State Circuits*, vol. 48, pp. 1453–1464, Jun 2013.

[85] C. Kun, A. Mason, and S. Chakrabartty, "A Dynamic Reconfigurable A/D Converter for Sensor Applications," in *Proceedings Paper 4th IEEE Conference on Sensors*, vol. 1 & 2 of *2005 IEEE Sensors*, (Irvine, CA, US), IEEE, Oct 2005.

[86] X. Jin, Z. Liu, and J. Yang, "New Flash ADC Scheme With Maximal 13 Bit Variable Resolution and Reduced Clipped Noise for High-Performance Imaging Sensor," *IEEE Sensors Journal*, vol. 13, pp. 167–171, Jan 2013.

[87] J. M. Dias Pereira, O. Postolache, and P. M. B. Silva Girao, "A Digitally Programmable A/D Converter for Smart Sensors Applications," *IEEE Transactions on Instrumentation and Measurement*, vol. 56, pp. 158–163, Feb 2007.

[88] S. Toscher, T. Reinemann, R. Kasper, and M. Hartmann, "A Reconfigurable $\Delta\Sigma$ ADC," in *IEEE International Symposium on Industrial Electronics (ISIE'06)*, vol. 1, pp. 495–499, Jul 2006.

[89] S.-Y. Lee and C.-J. Cheng, "A Low-Voltage and Low-Power Adaptive Switched-Current Sigma Delta ADC for Bio-Acquisition Microsystems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, pp. 2628–2636, Dec 2006.

[90] W. Qu, *Design of a Cost-efficient Reconfigurable Pipeline ADC*. PhD Dissertation, The University of Tennessee, Knoxville, US, Dec 2007.

[91] A. Rodriguez-Perez, M. Delgado-Restituto, and F. Medeiro, "A Low-Voltage Low-Power Successive Approximation Reconfigurable ADC Based on SC Techniques," in *Ph.D. Research in Microelectronics and Electronics (PRIME'09)*, pp. 8–11, Jul 2009.

[92] Y. Gao and H. Wang, "A Reconfigurable ADC Circuit with Online-Testing Capability and Enhanced Fault Tolerance," in *24th IEEE International*

Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'09), pp. 202–210, Oct 2009.

[93] H. Zhang, J. Tan, C. Zhang, H. Chen, and E. Sanchez-Sinencio, "A 0.6-to-200MSPS Speed Reconfigurable and 1.9-to-27mW Power Scalable 10-bit ADC," in *Proceedings of the 37th European Solid-State Circuits Conference (ESSCIRC'11)*, pp. 367–370, Sep 2011.

[94] S. Danesh, J. Hurwitz, K. Findlater, D. Renshaw, and R. Henderson, "A Reconfigurable 1 GSps to 250 MSps, 7-bit to 9-bit Highly Time-Interleaved Counter ADC with Low Power Comparator Design," *IEEE Journal of Solid-State Circuits*, vol. 48, pp. 733–748, Mar 2013.

[95] J. Risques, J. Duarte, V. Amaro, S. P. U, K. V. Chiang, K. F. Chang, and K. C. Lai, "A Very Area/Power Efficient Mixed Signal Circuit for Voice Signal Processing in 0.18 Digital Technology," in *Proceedings of the 29th European Solid-State Circuits Conference (ESSCIRC'03)*, pp. 169–172, Sep 2003.

[96] K. Masselos, A. Pelkonen, M. Cupák, and S. Blionas, "Realization of Wireless Multimedia Communication Systems on Reconfigurable Platforms," *EUROMICRO Journal of Systems Architecture, Special Issue: Reconfigurable Systems*, vol. 49, pp. 155–175, Sep 2003.

[97] A. Rusu, D. Rodriguez de Llera Gonzalez, and M. Ismail, "Reconfigurable ADCs Enable Smart Radios for 4G Wireless Connectivity," *IEEE Circuits and Devices Magazine*, vol. 22, pp. 6–11, Jun 2006.

[98] Y. Ke, J. Craninckx, and G. Gielen, "A Design Approach for Power-Optimized Fully Reconfigurable $\Delta\Sigma$ A/D Converter for 4G Radios," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 55, pp. 229 – 233, Mar 2008.

[99] Y. Wei, A. Doboli, and H. Tang, "Systematic Methodology for Designing Reconfigurable $\Delta\Sigma$ Modulator Topologies for Multimode Communication Systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, pp. 480–496, Mar 2007.

[100] R. Veljanovski, A. Stojcevski, J. Singh, A. Zayegh, and M. Faulkner, "Reconfigurable Architecture for UTRA-TDD System," *Electronics Letters*, vol. 38, pp. 1732–1733, Dec 2002.

[101] B. Xia, A. Valdes-Garcia, and E. Sanchez-Sinencio, "A 10-bit 44-MS/s 20-mW Configurable Time-Interleaved Pipeline ADC for a Dual-Mode 802.11b/Bluetooth Receiver," *IEEE Journal of Solid-State Circuits*, vol. 41, pp. 530–539, Mar 2006.

[102] Y. Ke, J. Craninckx, and G. Gielen, "A Design Methodology for Fully Reconfigurable $\Delta\Sigma$ Data Converters," in *Proceedings of Design, Automation and Test in Europe Conference & Exhibition (DATE'09)*, vol. 1 – 3, pp. 1379 – 1384, Nice, France: IEEE, Apr 2009.

[103] B. J. Farahani and M. Ismail, "Adaptive Sigma Delta ADC for WiMAX Fixed Point Wireless Applications," in *48th Midwest Symposium on Circuits and Systems (MWSCAS'05)*, vol. 1, (Cincinnati, OH, United States), pp. 692 – 695, Aug 2005.

[104] C. Jabbour, D. Camarero, V. T. Nguyen, and P. Loumeau, "A 1V 65nm CMOS Reconfigurable Time-Interleaved High-Pass $\Delta\Sigma$ ADC," in *IEEE Conference on Circuits and Systems (ISCAS'09)*, pp. 1557–1560, May 2009.

[105] C. C. Hsu, C. C. Huang, Y. H. Lin, C. C. Lee, Z. Soe, T. Aytur, and R. H. Yan, "A 7b 1.1GS/s Reconfigurable Time-Interleaved ADC in 90nm CMOS," in *Symposium on VLSI Circuits. Digest of Technical Papers.*, (Tokyo), Japan Society Applied Physics, Jun 2007.

[106] C. Lu, T. Xi, and M. Hao, "A Reconfigurable ΣΔ Modulator for Multi-Standard Wireless Application," in *9th International Conference on Solid-State and Integrated-Circuit Technology (ICSICT'08)*, pp. 1917–1920, Oct 2008.

[107] A. Silva, J. Guilherme, and N. Horta, "Reconfigurable Multi-Mode ΣΔ Modulator for 4G Mobile Terminals," *VLSI Journal on Integration*, vol. 42, pp. 34–46, Jan 2009.

[108] A. Silva, N. Horta, and J. Guilherme, "A Reconfigurable A/D Converter for 4G Wireless Systems," in *Proceedings of 2008 IEEE International Symposium on Circuits and Systems (ISCAS'08)*, pp. 924–927, Seattle: IEEE, May 2008.

[109] M. Taherzadeh-Sani and A. Hamoui, "A Reconfigurable and Power-Scalable 10-12 Bit 0.4-44 MS/s Pipelined ADC With 0.35-0.5 pJ/Step in 1.2V 90nm Digital CMOS," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, pp. 74–83, Jan 2013.

[110] T. Sun, N. Wang, Q. Zhang, D. Yuan, and H. Wang, "A Dynamic Power Programmable 10-bit 20-MS/s Pipeline ADC for ISM Band Wireless Communication," in *IEEE 11th International Conference on Solid-State and Integrated Circuit Technology (ICSICT'12)*, pp. 1–3, Oct 2012.

[111] J. Yoo, D. Lee, K. Choi, and J. Kim, "A Power and Resolution Adaptive Flash Analog-to-Digital Converter," in *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED'02)*, pp. 233–236, Aug 2002.

[112] J. Mallek, H. Mnif, and M. Loulou, "Flexible ΣΔ ADC for Mobile WiMAX Applications," in *16th IEEE Mediterranean Electrotechnical Conference (MELECON'12)*, pp. 502–505, Mar 2012.

[113] B. Xia, A. Valdes-Garcia, and E. Sanchez-Sinencio, "A Configurable Time-Interleaved Pipeline ADC for Multi-Standard Wireless Receivers," in *Proceeding of the 30th European Solid-State Circuits Conference (ESSCIRC'04)*, pp. 259–262, Sep 2004.

[114] P. Malla, H. Lakdawala, K. Kornegay, and K. Soumyanath, "A 28mW Spectrum-Sensing Reconfigurable 20MHz 72dB-SNR 70dB-SNDR DT $\Sigma\Delta$ ADC for 802.11n/WiMAX Receivers," in *IEEE International Solid-State Circuits Conference (ISSCC'08). Digest of Technical Papers.*, pp. 496–631, Feb 2008.

[115] M. Anderson, K. Norling, A. Dreyfert, and J. Yuan, "A Reconfigurable Pipelined ADC in 0.18μm CMOS," in *Proceedings 2005 IEEE Symposium on VLSI Circuits. Digest Technical Papers*, (Kyoto, Japan), pp. 326–329, Jun 2005.

[116] E. M. I. Gustafsson, A. Rusu, and M. Ismail, "Design of a Reconfigurable ADC for UWB/Bluetooth Radios," in *IEEE North-East Workshop on Circuits and Systems (NEWCAS'08)*, 2008 Joint IEEE North-East Workshop on Circuits and Systems and Taisa Conference, (New York), pp. 205 – 208, IEEE, Jun 2008.

[117] H.-K. Hong, H.-W. Kang, B. Sung, C.-H. Lee, M. Choi, H.-J. Park, and S.-T. Ryu, "An 8.6 ENOB 900MS/s Time-Interleaved 2b/cycle SAR ADC with a 1b/cycle Reconfiguration for Resolution Enhancement," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC'13)*, pp. 470–471, Feb 2013.

[118] B. Verbruggen, M. Iriguchi, and J. Craninckx, "A 1.7 mW 11b 250 MS/s 2-Times Interleaved Fully Dynamic Pipelined SAR ADC in 40nm Digital CMOS," *IEEE Journal of Solid-State Circuits*, vol. 47, pp. 2880–2887, Dec 2012.

[119] A. Stojcevski, J. Singh, and A. Zayegh, "Reconfigurable ADC for 3-G UTRA-TDD Mobile Receiver," in *2003 Southwest Symposium on Mixed-Signal Design*, pp. 27–31, Feb 2003.

[120] A. Stojcevski, J. Singh, and A. Zayegh, "Scalable Pipeline Analog-to-Digital Converter for UTRA-TDD Mobile Station Receiver," *Proceedings of IEEE International Conference on Semiconductor Electronics (ICSE'02)*, pp. 370–374, Dec 2002.

[121] A. Stojcevski, J. Singh, and A. Zayegh, "CMOS ADC with Reconfigurable Properties for a Cellular Handset," in *Proceedings of the Second IEEE International Workshop on Electronic Design, Test and Applications (DELTA'04)*, (DC, US), pp. 103 – 107, IEEE Computer Society, Jan 2004.

[122] B. J. Minnis and P. A. Moore, "A Highly Digitized Multimode Receiver Architecture for 3G Mobiles," *IEEE Transactions on Vehicular Technology*, pp. 637–653, May 2003.

[123] A. Stojcevski, V. Vibhute, J. Singh, and A. Zayegh, "Control Unit Implementation for a Reconfigurable ADC," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 5274, pp. 286 – 294, Mar 2004.

[124] A. Stojcevski, R. Singh, M. Faulkner, and A. Zayegh, "A Real-Time Reconfigurable Pipelined Architecture with Advanced Power Management for UTRA-FDD," *Proceedings of 14th IEEE 2003 International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'03)*, vol. 1, pp. 64 – 68, Sep 2003.

[125] S. Veeramachanen, A. M. Kumar, V. Tummala, and M. B. Srinivas, "Design of a Low Power, Variable-Resolution Flash ADC," in *22nd International Conference on VLSI Design*, pp. 117–122, Jan 2009.

[126] A. Stojcevski, J. Singh, and A. Zayegh, "Low Power, 3-bit CMOS Pipeline ADC with Reduced Complexity Flash Architecture," in *32th European*

Solid-State Device Research Conference (ESSDERC'02), (Firenze, Italy), Sep 2002.

[127] A. Stojcevski, H. Le, J. Singh, and A. Zayegh, "Flash ADC Architecture," *Electronics Letters*, vol. 39, pp. 501 – 502, Mar 2003.

[128] J. Chandran, R. Kaluri, J. Singh, V. Owall, and R. Veljanovski, "Xilinx Virtex II Pro Implementation of a Reconfigurable UMTS Digital Channel Filter," in *Second IEEE International Workshop on Electronic Design, Test and Applications (DELTA'04)*, pp. 77–82, Jan 2004.

[129] R. Veljanovski, J. Singh, and M. Faulkner, "DSP and ASIC Implementation of a Channel Filter for a 3G UTRA-TDD System," in *The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'02)*, vol. 3, pp. 1447–1451, Sep 2002.

[130] R. Veljanovski, J. Singh, and M. Faulkner, "FIR Pulse Shaping Filter for a 3G Wireless TDD Application," in *Proceedings of International Symposium on Integrated Circuits, Devices & Systems (ISIC'01)*, (Marina Mandarin, Singpore), Sep 2001.

[131] P. Lajevardi, A. P. Chandrakasan, and H.-S. Lee, "Zero-Crossing Detector Based Reconfigurable Analog System," *IEEE Journal of Solid-State Circuits*, vol. 46, pp. 2478–2487, Nov 2011.

[132] 3rd Generation Partnership Project (3GPP), "TS 25.101 Technical Specification Group Radio Access Networks: Base Station (BS) Radio Transmission and Reception - FDD," vol. 3.7.0, 2001.

[133] Nokia Mobile Phones, "MS Receiver Sensitivity in UTRA FDD Mode," Technical Report, TSG-RAN Working Group 4 (TSGW4), 3rd Generation Partnership Project (3GPP), Jan 1999.

[134] J. Jussila, *Analog Baseband Circuits for WCDMA Direct-Conversion Receivers*. PhD Dissertation, Electronic Circuit Design Laboratory, Department of Electrical and Communications Engineering, Helsinki University of Technology, Finland, Jun 2003.

[135] S. Reekmans, P. Rombouts, and L. Weyten, "Sigma Delta ADC Design Considerations for an UMTS Receiver," in *The 49th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS'06)*, vol. 2, Aug 2006.

[136] J.-H. Kim, C.-S. Chae, Y.-J. Woo, and G.-H. Cho, "A CMOS Variable Gain Amplifier with Wide Dynamic Range and Accurate dB-Linear Characteristic," in *the 8th International Conference Advanced Communication Technology (ICACT'06)*, vol. 1, pp. 831–835, Feb 2006.

[137] B. Razavi, "Design Considerations for Direct-Conversion Receivers," *IEEE Transactions on Circuits and Systems*, vol. 44, pp. 428–435, Jun 1997.

[138] 3rd Generation Partnership Project (3GPP), "TS 25.104 Technical Specification Group Radio Access Networks: BS Radio Transmission and Reception - FDD," vol. V4.5.0, 2002.

[139] 3rd Generation Partnership Project (3GPP), "NTT Technical Review: Trends in the Research and Development of Software Defined Radio," vol. 1, no. 4, pp. 10 – 14, 2003.

[140] B. Walke, P. Seidenberg, and M. P. Althoff, *UMTS: The Fundamentals*. John Wiley & Sons, 2002.

[141] P. Marques and A. Gameiro, "Sensing Opportunities in UMTS Spectrum," in *2nd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom'07)*, pp. 492 – 496, Aug 2007.

[142] P. Marques, J. Bastos, and A. Gameiro, "Sensing UMTS Bands Using Cyclostationary Features and Cooperation Between Opportunistic Terminals," in *3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom'08)*, pp. 1 – 5, May 2008.

[143] M. M. Öner, *Air Interface Identification for Software Radio Systems*. PhD Dissertation, Institute of Communication System, University of Karlsruhe, Germany, Dec 2004.

[144] M. M. Öner and F. Jondral, "Air Interface Identification for Software Radio Systems," {*AEU*} - *International Journal of Electronics and Communications*, vol. 61, pp. 104 – 117, Feb 2007.

[145] Y. Pershin, E. Sazonov, and M. Ventra, "Analogue-to-Digital and Digital-to-Analogue Conversion with Memristive Devices," *Electronics Letters*, vol. 48, pp. 73 – 74, Jan 2012.

[146] R. Kline, "Harold Black and the Negative-Feedback Amplifier," *IEEE Control Systems*, vol. 13, pp. 82 – 85, Aug 1993.

[147] K. Ogata, *Modern Control Engineering*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 4th ed., 2001.

[148] I. Brzozowski and A. Kos, "Minimization of Power Consumption in Digital Integrated Circuits by Reduction of Switching Activity," in *Proceedings of 25th EUROMICRO Conference*, vol. 1, pp. 1376 – 1380, Sep 1999.

[149] R. V. Menon, S. Chennupati, N. K. Samala, D. Radhakrisnan, and B. Izadi, "Minimization Switching Activity Minization in Combinational Logic Design," in *Proceedings of International Conference on Embedded Systems and Applications*, pp. 47 – 53, Jun 2004.

[150] R. Ferreira and A. Trullemans, "An Efficient Switching Activity Estimation for Low Power Resynthesis Under General Delay," in *PATMOS International Workshop*, pp. 18–19, Oct 1999.

[151] T. English, K. L. Man, and E. Popovici, "A Switching Activity Analysis and Visualisation Tool for Power Optimisation of SoC Buses," in *Ph.D. Research in Microelectronics and Electronics (PRIME'09)*, pp. 264 – 267, Jul 2009.

[152] F. N. Najm, "Transition Density: A New Measure of Activity in Digital Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, pp. 310 – 323, Feb 1993.

[153] C. Baena, J. Juan-Chico, M. J. Bellido, P. R. d. Clavijo, C. J. Jiménez, and M. Valencia, "Measurement of the Switching Activity of CMOS Digital Circuits at the Gate Level," in *Proceedings of the 12th International Workshop on Integrated Circuit Design. Power and Timing Modeling, Optimization and Simulation (PATMOS'02)*, (London, UK), pp. 353 – 362, Springer-Verlag, Sep 2002.

[154] S. Bhanja and N. Ranganathan, "Modeling Switching Activity Using Cascaded Bayesian Networks for Correlated Input Streams," in *Proceedings of IEEE International Conference on Computer Design: VLSI in Computers and Processors (ICCD'02)*, pp. 388 – 390, Sep 2002.

[155] F. A. Aloul and A. Sagahyroon, "Estimation of the Weighted Maximum Switching Activity in Combinational CMOS Circuits," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS'06)*, pp. 2929 – 2932, May 2006.

[156] R. W. Hamming, "Error Detecting and Error Correcting Codes," *Bell System Technical Journal*, pp. 147 – 160, Apr 1950.

[157] S. Johansson, S. He, and P. Nilsson, "Wordlength Optimization of a Pipelined FFT Processor," in *Proceedings of 42nd IEEE Midwest Sympo-*

sium on Circuits and Systems (MWSCAS'99), vol. 1, pp. 501–503, Aug 1999.

[158] S. Lewis and P. Gray, "A Pipelined 5MHz 9b ADC," in *1987 IEEE International Solid-State Circuits Conference. Digest of Technical Papers.*, vol. XXX, pp. 210–211, Feb 1987.

[159] S. Lewis and P. Gray, "A Pipelined 5-Msample/s 9-bit Analog-to-Digital Converter," *IEEE Journal of Solid-State Circuits*, vol. 22, pp. 954–961, Dec 1987.

[160] C. Conroy, D. Cline, and P. Gray, "An 8-b 85-MS/s Parallel Pipeline A/D Converter in 1μm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 28, pp. 447–454, Apr 1993.

[161] J. M. Rabaey and M. Pedram, *Low Power Design Methodologies*. Kluwer Academic Publishers, 1996.

[162] A. Chandrakasan, S. Sheng, and R. Brodersen, "Low-Power CMOS Digital Design," *IEEE Journal of Solid-State Circuits*, vol. 27, pp. 473 – 484, Apr 1992.

[163] A. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter Control in Evolutionary Algorithms," vol. 3, pp. 124 – 141, Jul 1999.

[164] M. Mitchell, *An Introduction to Genetic Algorithms*. The MIT Pres, 2002.

[165] W. Audoglio, E. Zuffetti, G. Cesura, and R. Castello, "A 6-10 bits Reconfigurable 20MS/s Digitally Enhanced Pipelined ADC for Multi-Standard Wireless Terminals," in *Proceedings of the 32nd European Solid-State Circuits Conference (ESSCIRC'06)*, (Montreux, Switzerland), pp. 496–499, Sep 2006.

[166] R. Veljanovski, J. Singh, and M. Faulkner, "Digital Control Unit Implementation for a Reconfigurable ADC," in *10th International Symposium on Integrated Circuits, Devices & Systems (ISIC'04)*, (Singapore), Sep 2004.

[167] I. Ahmed and D. A. Johns, "A 50-MS/s ($35mW$) to 1-kS/s ($15\mu W$) Power Scaleable 10-bit Pipelined ADC using Rapid Power-On Opamps and Minimal Bias Current Variation," *IEEE Journal of Solid-State Circuits*, vol. 40, pp. 2446–2455, Dec 2005.

[168] K. Iizuka, H. Matsui, M. Ueda, and M. Daito, "A 14-bit Digitally Self-Calibrated Pipelined ADC with Adaptive Bias Optimization for Arbitrary Speeds up to 40 MS/s," *IEEE Journal of Solid-State Circuits*, vol. 41, pp. 883–890, Apr 2006.

[169] K. Chandrashekar, M. Corsi, J. Fattaruso, and B. Bakkaloglu, "A 20-MS/s to 40-MS/s Reconfigurable Pipeline ADC Implemented With Parallel OTA Scaling," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, pp. 602 – 606, Aug 2010.

[170] H. Voorman and H. Veenstra, "Tunable High-Frequency $G_m - C$ Filters," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 8, pp. 1097 – 1108, 2000.

[171] E. G. Friedman, "Substrate Coupling and Interconnect Noise in Mixed-Signal and High speed Digital ICs," in *IEEE CAS Workshop on Mixed-Signal Integrated Circuit Design*, Dec 1999.

[172] R. Rossi, G. Torelli, and V. Liberali, "Model and Verification of Triple-Well Shielding on Substrate Noise in Mixed-Signal CMOS ICs," in *Proceedings of the 29th European Solid-State Circuits Conference (ESSCIRC'03)*, pp. 643 – 646, Sep 2003.

[173] X. Aragones, J. L. Gonzalez, and A. Rubio, *Analysis and Solutions for Switching Noise Coupling in Mixed-Signal ICs*. Kluwer Academic Publishers, 1999.

[174] X. Aragones, F. Moll, M. Roca, and A. Rubio, "Analysis and Modelling of Parasitic Substrate Coupling in CMOS Circuits," *IEE Proceedings of Circuits, Devices and Systems*, vol. 142, pp. 307 – 312, Oct 1995.

[175] G. Vasilescu, *Electronic Noise and Interfering Signals: Principle and Applications*. Berlin: Springer, 1995.

[176] L. W. Couch II, *Modern Communication Systems: Principle and Applications*. Prentice Hall, 1995.

[177] R. McDonough and A. Whalen, *Detection of Signals in Noise*. Academic Press, 1995.

[178] S. A. Klein and D. M. Levi, "Stochastic Model for Detection of Signals in Noise," *Journal of the Optical Society of America A*, vol. 26, pp. B110 – B126, Nov 2009.

[179] K.-H. Lee, S.-W. Lee, Y.-J. Kim, K.-S. Kim, and S.-H. Lee, "Ten-bit 100 MS/s 24.2 mW $0.8mm^2$ $0.18\mu m$ CMOS Pipeline ADC Based on Maximal Circuit Sharing Schemes," *Electronics Letters*, vol. 45, pp. 1296 – 1297, Dec 2009.

[180] B.-S. Park, S.-H. Ji, M.-H. Choi, K.-H. Lee, G.-C. Ahn, and S.-H. Lee, "A 10b 100MS/s 25.2mW $0.18\mu m$ CMOS ADC with Various Circuit Sharing Techniques," in *International SoC Design Conference (ISOCC'09)*, pp. 329 – 332, Nov 2009.

[181] J. Li and U.-K. Moon, "A 1.8-V 67-mW 10-bit 100-MS/s Pipelined ADC Using Time-Shifted CDS Technique," *IEEE Journal of Solid-State Circuits*, vol. 39, pp. 1468 – 1476, Sep 2004.

[182] H.-Y. Lee and S.-I. Liu, "A 10-bit 100MS/s Pipelined ADC in $0.18\mu m$ CMOS Technology," in *IEEE International SOC Conference (SOCC'07)*, pp. 3 – 6, Sep 2007.

[183] Y.-I. Park, S. Karthikeyan, F. Tsay, and E. Bartolome, "A 10-b 100 MSample/s CMOS Pipelined ADC with 1.8V Power Supply," in *IEEE International Solid-State Circuits Conference (ISSCC'01). Digest of Technical Papers*, pp. 130 – 131, Feb 2001.